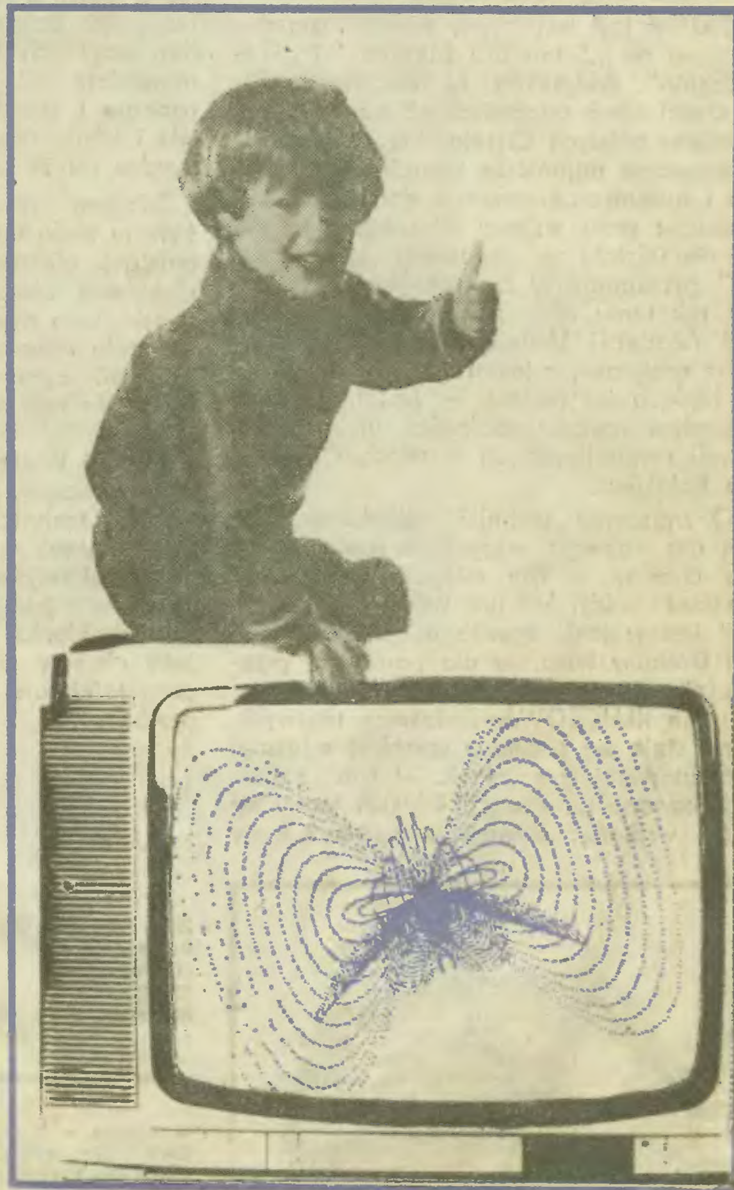
bajtek

Z MIKROKOMPUTEREM NA TY

NR 1 Cena 60 zł

WRZESIEŃ — 85

- Prof. Władysław M. Turski zaprasza
- ZX Spectrum
- LOGO
- ATIC—ATAC
- Ochrona programów
- Drążek sterowy (joystick)
- Mikrokomputery na świecie



RUN czyli zaczynamy

BAJT – to 8 bitów, podstawowych jednostek informacji.

BAJTEK – to popularne pismo, poświęcone temu wszystkiemu, co z przekształcaniem informacji się wiąże. Przede wszystkim chcemy być pomocni tym, którzy już mają lub chcą mieć komputery osobiste.

BAJTEK jest wspólnym, miesięcznym dodatkiem do „Sztandaru Młodych" i „Odrodzenia". Podjęliśmy tę inicjatywę, aby w części choć odpowiedzieć na zapotrzebowanie młodych Czytelników. Że to zainteresowanie najnowszą techniką jest wielkie i autentyczne, poparte dodatkowo desperacką wolą wzięcia własnego, liczącego się udziału w światowej „grze o jutro" przekonaliśmy się raz jeszcze rzucając rok temu ideę powołania Młodzieżowej Akademii Umiejętności. Wielki rezonans społeczny, z jakim spotkała się MAU – będąca już faktem – jest najlepszym dowodem ogromu zdolności, umiejętności i woli czynu tkwiących w młodym pokoleniu Polaków.

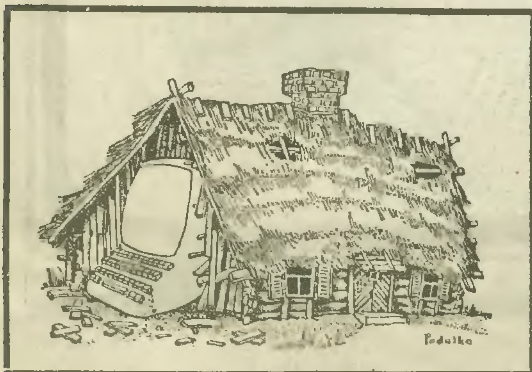
O znaczeniu techniki mikrokomputerowej dla rozwoju wszystkich społeczeństw nie chcemy w tym miejscu pisać, gdyż przecież każdy, kto już wziął BAJTKA do ręki jest w jakiś sposób o tym przekonany! Dodamy tylko, że dla pomyślnej przyszłości naszego kraju jest to sprawa absolutnie KLUCZOWA. Zwłaszcza niezwykle pilne staje się podjęcie szerokiej edukacji komputerowej w szkołach. W tym sensie publikowane na łamach BAJTKA materiały będą wspierać zainicjowany przez Radę Krajową PRON „Narodowy Czyn Pomocy Szkole".

Ambicją zespołu redakcyjnego jest – najogólniej mówiąc – zwalczanie analfabetyzmu mikrokomputerowego w Polsce. Oczekujemy pomocy tych wszystkich, którzy w walce o sprawy informatyki w Polsce mają już doświadczenie. Nasze łamy są dla wszystkich szeroko otwarte. Zaproponowaliśmy już zresztą współpracę wielu osobom i środowiskom twórców informatyki i użytkownikom jej metod i urządzeń. Bardzo na tę pomoc liczymy!

Zdajemy sobie sprawę, że na całym świecie wydawnictwa tego typu są o wiele bardziej efektowne. Kolor, dobry papier, efektowne okładki, wkładki, kasety i dyskietki, jako niezbędne załączniki – to już się stało normą. Żeby nie szukać daleko: zazdrość ogarnia, gdy się na przykład bierze do ręki tego typu pisma oferowane młodym Bułgarom! Mamy nadzieję, że wkrótce i BAJTEK uzyska odpowiednie do roli społecznej, jaką chcemy żeby spełniał, warunki techniczne. Tymczasem – zaczynamy jednak w takim kształcie, w jakim to jest aktualnie możliwe, gdyż strata każdego dnia jest w tej dziedzinie stratą nie do odrobienia. Alternatywy nie mamy – jeśli chcemy nie tylko marzyć o lepszej przyszłości, ale i praktycznie tę przyszłośc przybliżać.

Z-ca redaktora naczelnego „Odrodzenia"
ZBIGNIEW SIEDLECKI

Z-ca redaktora naczelnego „Sztandaru Młodych"
WALDEMAR SIWIŃSKI



Wydawca — Krajowe Wydawnictwo Czasopism RSW „Prasa-Książka-Ruch" ul. Noakowskiego 14, 00-666 Warszawa, tel. 257294.
Dodatek specjalny „Odrodzenia" i „Sztandaru Młodych". Cena 60 zł.

Zam. 1922. N-23



Rozmowa z prof. WŁADYSŁAWEM M. TURSKIM — informatykiem, profesorem Uniwersytetu Warszawskiego i Uniwersytetu Londyńskiego, prezesem Polskiego Towarzystwa Informatycznego.

Matchbox dla szofera

● Na początek poinformujemy czytelników, iż umówiliśmy się rozmawiać z Panem Profesorem jako z autorem głośnej książki „Nie samą informatyką". Będziemy, jednym słowem, rozliczać Pana Profesora z publicznych wypowiedzi. W swojej książce sformułował Pan Profesor przestrogę, że niezajmowanie się informatyką grozi kolonizacją cywilizacyjną i intelektualną. Jak widzi Pan ten problem w chwili obecnej?

— Sześć lat po napisaniu książki powtarzam przestrogę i ze smutkiem stwierdzam, że mimo fali publicznego lecz bardzo powierzchownego zainteresowania sprawami informatyki (właściwie jej widowiskową stroną), zagrożenie nie tylko występuje nadal, ale pogłębiło się o te sześć lat, podczas których nie zrobiono nic, aby sprawę zastosowań informatyki w Polsce postawić na zdrowszych podstawach niż dotychczas.

Miały miejsce w tych latach zjawiska bardzo smutne. Niewątpliwie straciliśmy sporo kadry, zarówno przez wyjazdy zagraniczne, jak i przez odejście z zawodu. Niech nie łudzą się ci, którzy sądzą, że można by temu przeciwdziałać metodami administracyjnymi np. przez ograniczenia w wydawaniu paszportów informatykom. Nie zapobiegnie to odpływowi tych informatyków, którym bardziej opłaca się naprawa telewizorów, czy wręcz prowadzenie butików.

Ten proces, niestety, trwa nadal; towarzyszy mu (i w dużym stopniu go wywołuje) zjawisko dalszej archaizacji wyposażenia uczelni w sprzęt informatyczny. Od ukazania się wspomnianej książki już pięć roczników absolwentów szkół ekonomicznych i technicznych (tj. tych, którzy najbardziej do używania informatyki powinni być wdrożeni) opuściło mury uczelni, nie mając zielonego pojęcia jak tę informatykę naprawdę stosować. Nie budzi to mojego entuzjazmu.

O ile jeszcze pięć lat temu, w tych kilku instytutach informatyki, które traktują sprawę poważnie, kształciliśmy ludzi na mniej więcej takim samym poziomie przygotowania co przeciętne uczelnie zagraniczne, to w tej chwili jest to już zupełnie niemożliwe. Nie posiadamy nie tylko równorzędnych, czy trochę tylko gorszych, lecz w ogóle żadnych urządzeń pozwalających kształcić np. w grafice komputerowej czy nowoczesnych, ergonomicznych systemach dostępu do komputerów. W wielu ważnych gałęziach informatyki kształcenie odbywa się na zasadzie opowiadania jak to wygląda.

● W wywiadzie udzielonym „Polityce" stwierdził Pan Profesor, że nawet ci informatycy, którzy wyjechali za granicę powróciliby do kraju, gdyby zaistniały ku temu sprzyjające warunki. Co więc należy robić w tym celu?

— Sądzę, że wielu z nich powróciłoby. Niezbędne są dwa warunki: oczywisty, który wymienię na drugim miejscu i mniej oczywisty, od którego zacznę. Muszą oni poczuć się rzeczywiście potrzebni. Jestem przekonany, że otwarcie realnych perspektyw zawodowych w kraju będzie bardzo silnym bodźcem. Rzeczywiste zapotrzebowanie na informatyków może jednak stworzyć tylko poważny, ogólnonarodowy program rozwoju zastosowań informatyki, uwzględniający wszystkie uwarunkowania techniczne, ekonomiczne i społeczne.

Teraz ów warunek oczywisty, czyli sprawa płac. Nie ma co ukrywać, że informatycy są świadomi tego, jaką obiektywną wartość przedstawia ich wysoko kwalifikowana praca na rynku światowym. Dysproporcje w honorowaniu ich pracy w Polsce i za granicą, ostrzejsze bodajże niż w jakimkolwiek innym zawodzie, stanowią dla większości młodych ludzi argument nie tylko ekonomiczny. Jest to widomy znak społecznego niedocenienia.

● Trzy lata temu w wywiadzie dla „ITD" stwierdził Pan: „Nie chcę tylko trwać. Mnie interesuje podjęcie agresywnego rozwoju od dziś, od zaraz". Czy pozostał Pan wierny tej postawie?

— Trudno odpowiedzieć na to pytanie w kategoriach, w których umówiliśmy się prowadzić tę rozmowę, to znaczy — autorskich. Osobiście nie mam poczucia zmarnowania tych lat, ani stępienia własnej agresywności.

Uważam, że bardzo wiele udało się osiągnąć zespołowi ludzi, którzy tworzą Polskie Towarzystwo Informatyczne. Fakt systematycznego wzrostu liczbowego Towarzystwa, które nadal utrzymuje dość ostre kryteria akceptacji nowych członków, dobrze zorganizowane i cieszące się dużą popularnością imprezy kształceniowe, prowadzenie działalności szkoleniowo-badawczej na zlecenie wielu instytucji, funkcjonowanie Towarzystwa bez pobierania żadnych zasiłków, a równocześnie finansowanie wielu ciekawych akcji (między innymi — kupowanie komputerów dla szkolnych kół zainteresowań), nawiązanie rzeczowej współpracy z resortem oświaty i dobrych kontaktów ze środowiskiem nauczycielskim — wszystko to świadczy o skutecznej aktywności informatyków, zwłaszcza na tle ogólnego przekonania, że w Polsce niewiele można zrobić społecznym wysiłkiem.

● W swojej książce niewiele miejsca poświęcił Pan zjawiskom powszechnego zainteresowania informatyką. Jak ocenia Pan istotność tego zjawiska?

— W dwóch rozdziałach mojej książki duże ustępy poświęcone są masowości informatyki, nie ma natomiast niczego o zjawisku popkultury informatycznej, ponieważ tego zjawiska po prostu nie było. Pisałem o innej masowości, o komputerach domowych połączonych siecią. Rozwija się to znacznie wolniej niż przypuszczałem, a w Polsce nie występuje w ogóle.

Nasze domowe komputery, działając jako urządzenia do przetwarzania napisów, redagowania tekstów są z pewnością bardzo użyteczne dla ludzi zawodowo parających się „piórem" Sam z tego korzystam i bez mego manipulatora napisów czułbym się już jak bez ręki. Dotyczy to jednak bardzo nikłego procentu społeczeństwa. Dla większości ludzi mikrokomputery odizolowane od dużych banków informacji nie spełniają (poza rozrywką) żadnych określonych funkcji.

Nie można nie doceniać rozrywki, nie wolno jej jednak przeceniać, zwłaszcza że „zdolności rozrywkowe" taniego mikrokomputerka są niewielkie i zabawa w zasadzie kończy się w momencie opanowania tego sprzętu. Meteoryczna kariera najprostszych urządzeń tego typu zdecydowanie przygasa na światowym horyzoncie.

Prywatny komputer to przede wszystkim informator inteligentny, selektywny, choć pamiętający o wszystkim, spontaniczny i niestrudzony. Musi jednak być zrealizowana funkcja łatwego dostępu do informacji nie generowanej na miejscu. Mówi się o wszelkiego rodzaju poradnikach, programach edukacyjnych do indywidualnego użytku; z całą pewnością jest to droga słuszna, lecz wymaga zawrotnej ilości doskonałego oprogramowania. Zróżnicowany rynek indywidualnych użytkowników nie jest w stanie utrzymać ciężaru kosztów jego przygotowania.

Proces masowego użytkowania izolowanych komputerów osobistych, po przejściu przez fazę początkowej fascynacji — jak sądzę — zamiera. Nie dotyczy to oczywiście personalnych komputerów zawodowych: biurowych, do redagowania tekstów, do prowadzenia księgowości (lecz jeśli ta księgowość ma obejmować jedynie budżet jednej rodziny to naprawdę kajecik w kratkę działa lepiej).

● A jak ocenia Pan Profesor tę modę w naszym kraju?

— Boję się ogromnie, że fala popkultury informatycznej zaczyna ludziom przysłaniać rzeczywistą informatykę. Dla większości ludzi w

DOKOŃCZENIE NA STR. 26

Co to jest mikrokomputer

> *„Wyśledzić moment historyczny, w którym liczydło dosięgło Rozumu, jest równie trudno, jak ów, co małpę przemienił w człowieka".*
>
> *Stanisław Lem „GOLEM XIV"*

Mikrokomputer składa się z pięciu najważniejszych dla użytkownika elementów: klawiatury, mikroprocesora, pamięci typu **ROM**, pamięci typu **RAM** i interfejsu. (rys. 1)

Zbiór linii sygnałowych, którymi przesyłane są dane między mikroprocesorem a pamięciami (ROM, RAM) i urządzeniami zewnętrznymi (poprzez interfejs I/O), nazywa się magistralą danych. Ponieważ komunikacja jest dwustronna więc magistrala nazywa się dwukierunkowa Komórki pamięci oraz urządzenia zewnętrzne muszą być wybierane przez mikroprocesor selektywnie. przez podanie ich adresu liniami adresowymi. Zbiór linii adresowych nazywa się magistralą adresową. Jest to magistrala jednokierunkowa zn. adres jest przesyłany od mikroprocesora do pozostałych podzespołów. Ponadto w mikroprocesorze istnieje specjalny układ sterowania, który dekoduje rozkazy i generuje wewnętrzne sygnały sterujące, zapewniające właściwy przebieg operacji zdefiniowanej kodem rozkazu. Sygnały te, wspólnie z sygnałami wprowadzanymi do układu sterowania z urządzeń zewnętrznych przesyłane są tzw. magistralą sterującą.

MIKROPROCESOR

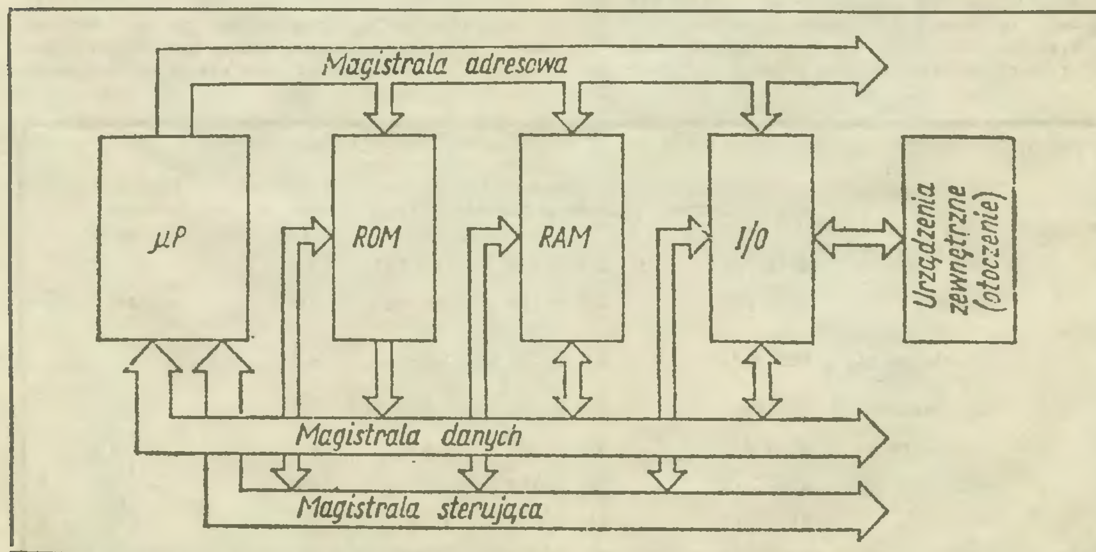
Zasadniczym i najbardziej skomplikowanym elementem mikrokomputera jest mikroprocesor **CPU** (z ang. Central Proccesing Unit). Jeśli jednocześnie przetwarza się ośmiobitowe (bit to 0 lub 1) porcje informacji to nazywa się 8-bitowym, w przypadku gdy porcje te są większe i wynoszą 16, 32 lub 64 bity mówi się o mikroprocesorach odpowiednio 16-, 32- i 64-bitowych. Oczywiście wraz ze wzrostem wielkości przetwarzanej informacji rośnie szybkość pracy mikroprocesora.

Mikroprocesor może operować na pamięci operacyjnej i na rejestrach czyli pamięci umieszczonej w samym mikroprocesorze. Można wyróżnić 8 podstawowych typów rozkazów:

— Ładowanie i modyfikacja zawartości pamięci i rejestrów.
— Przesyłanie i przeszukiwanie obszarów pamięci.
— Operacje arytmetyczne i logiczne.
— Rotacja i cyrkulacja zawartości komórek pamięci i rejestrów.
— Ustawianie poszczególnych bitów.
— Skoki, wywoływanie i powroty z podprogramów.
— Obsługa wejścia—wyjścia.
— Sterowanie stanem mikroprocesora.

Rozkaz jest wykonywany po przesłaniu do mikroprocesora informacji w postaci ciągu zer i jedynek, będącej kodem żądanego zlecenia. Każdemu rozkazowi odpowiada inny kod zero-jedynkowy, zrozumiały dla mikroprocesora.

Wykonywanie każdego rozkazu wymaga od 1 do 5 cykli mikroprocesora. Każdy cykl składa się z 3—5 taktów wbudowanego zegara. Częstotliwość zegara jest najważniejszym po architekturze czynnikiem wpływającym na szybkość wykonywanych operacji.



Schemat blokowy komputera

Dwa najbardziej rozpowszechnione w Polsce mikrokomputery Sinclair Spectrum i Commodore 64 zbudowane są na dwóch różnych mikroprocesorach, pierwszy na **Z80** firmy **ZILOG**, a drugi na **6510** (zgodny programowo z **6502** firmy **MOS Technology)**. Polski mikrokomputer **Compan-8**, wytwarzany w zabrzańskim **ZUK** ma konstrukcję opartą na **Intelu 8080A**, a mikrokomputer **Meritum 1** pracuje na mikroprocesorze **U880D** — NRD-owskim odpowiedniku **Z80**.

PAMIĘĆ

Sam mikroprocesor to jednak za mało, gdyż pracować może dopiero po przyłączeniu do niego pamięci. Zgodnie z intuicją, pamięć jest urządzeniem służącym do przechowywania informacji cyfrowych. Najprostszym modelem takiej pamięci jest rząd latarni wzdłuż wąskiej uliczki. Należy założyć, jak to zresztą często bywa, że nie wszystkie latarnie muszą się palić jednocześnie i to, że jeśli pewna latarnia się pali, to nie ma żadnego wpływu na inne. Każda latarnia jest teraz pojedynczą komórką i może znajdować się w dwóch stanach — zgaszona = 0 lub zapalona = 1. O takiej pamięci mówi się, że ma **organizację bitową.** Słowo **bit** oznacza cyfrę dwójkową tzn. 0 lub 1 (zgaszona lub zapalona latarnia).

W mikrokomputerach wyróżnia się pamięć **wewnętrzną** (operacyjną) i **zewnętrzną.** Pamięć operacyjna zbudowana jest z elemenetów półprzewodnikowych i służy do przechowywania danych i programów używanych na bieżąco przez mikroprocesor. Każdy mikrokomputer ma dwa rodzaje pamięci operacyjnej : pierwsza przeznaczona jest wyłącznie do czytania tzn. pobierania z niej informacji i nazywa się pamięcią stałą lub **ROM** (z ang. Read Only Memory), druga zapisywalno-odczytywalna daje możliwość dodatkowo zapisu informacji i nazywa się **RAM** (z ang. Random Access Memory).

Pamięć **ROM** zawiera stałe programy i dane, które umożliwiają mikroprocesorowi rozpoczęcie pracy natychmiast po włączeniu zasilania. Zawartość **ROM-u** jest niezmienna i nie do skasowania. Natomiast zawartość pamięci **RAM** jest tracona w momencie odłączenia zasilania.

Ilość pojedynczych komórek pamięci określa jej pojemność. Jeżeli przez 1 **bit** rozumie się pojemność jednej komórki, to **1 bajt** oznacza pojemność 8 komórek pamięci czyli 8 bitów. Pojemność pamięci przyjęło się wyrażać w **bajtach** lub **kilobajtach**, jednostkach 1024 razy większych niż 1 bajt.

Pojemność RAM-u większości średniej klasy mikrokomputerów wynosi do 64 kB tzn. 64 x 1024 bajty, czyli 64 x 1024 x 8 = 524288, to dużo, czy mało? Aby wyobrazić sobie co oznacza 64 kB, trzeba wiedzieć, że zapis kodu każdego znaku alfanumerycznego zajmuje 8 bitów czyli jeden bajt pamięci. Tak samo zapisanie dowolnej liczby od 0 do 255 wymaga jednego bajtu pamięci. To oznacza, że można zapisać w pamięci około 65 tysięcy liter lub liczb mniejszych od 255. Przeciętna strona maszynopisu tekstu lub książki średniego formatu zawiera 2 tysiące znaków, czyli w pamięci pojemności 64 kB można zapisać 32,5 strony maszynopisu lub 65 tysięcy liczb z przedziału 0—255.

PERYFERIA

Aby mikrokomputer stał się przydatny do pracy musi mieć możliwość komunikowania się z otoczeniem. Powyższe zadanie realizowane jest przez tzw. urządzenia zewnętrzne lub peryferyjne. Wszystkie urządzenia zewnętrzne łączone są z mikroprocesorem przez specjalny przyrząd nazywany **układem wejścia-wyjścia** lub **interfejsem.**

Mikrokomputer może współpracować z następującymi urządzeniami:
— magnetofonem kasetowym lub szpulowym
— wzmacniaczem i głośnikami
— monitorem lub telewizorem kolorowym i czarno-białym
— stacją dysków elastycznych
— drukarką lub maszyną do pisania
— modemem.

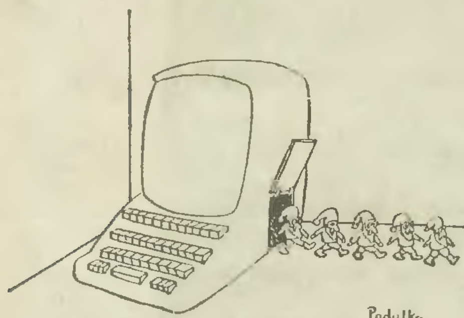
Monitor, telewizor i **głośniki** są urządzeniami zewnętrznymi, wyjściowymi, których mikrokomputer używa do przekazywania użytkownikowi informacji.

Magnetofon jest urządzeniem zarówno wejściowym jak i wyjściowym, które nie służy do bezpośredniej komunikacji, ale do przechowywania

Popularne mikrokomputery

mikrokomputer	mikroprocesor typ /l.bitów/	zd.rozdzielacza pixels x pixels	RAM kB	ROM kB	liczba kolorów
IBM	8088 /16/	320 x 300	64-512	40	16
Apple II	6502 /8/	256 x 192	48-112	16	8
Sinclaire QL	68008 /16/	512 x 256	128-640	48	4
ZX Spectrum 48	Z80 /8/	256 x 192	48	16	8
Commodore 64	6510 /8/	320 x 200	64	20	16
Atari 800XL	6502C /8/	320 x 192	64	24	256
Macintosh	68000 /16/	512 x 342	128-192		2
TRS-80 typ 2000	80186 /16	640 x 400	128-768		16
Schneider	Z80 /8/	640 x 200	64	32	27

programów i danych, **taśma magnetofonowa** jest nośnikiem informacji i spełnia rolę **pamięci zewnętrznej**. Analogiczna jest rola **dyskietki elastycznej** i **stacji dysków**, czyli **„drivera"**. Pojemność pamięci z jednej strony dyskietki wynosi około 170 kB, (dla Commodore 64) czyli około 100 stron maszynopisu.



Informacja może zostać również utrwalona na papierze, dzięki wykorzystaniu do tego celu **drukarki** lub **maszyny do pisania**. Aby zapewnić możliwość kontaktu dwóch mikrokomputerów, których nie można połączyć jednym przewodem stosuje się **modem** — urządzenie wykorzystujące linię telefoniczną do uzyskiwania bądź przesyłania informacji.

JĘZYKI

Wykonanie każdego zadania polega na zrealizowaniu pewnej sekwencji rozkazów nazywanej programem. Czy zatem znajomość listy rozkazów dla danego typu mikroprocesora, czyli jego języka wewnętrznego, jest niezbędnym warunkiem korzystania z mikrokomputera? Na szczęście nie. Istnieje wiele tzw. **języków programowania wyższego poziomu**, umożliwiających precyzowanie działań „maszyny" w znacznie bardziej syntetycznej postaci i zapisywanie ich w sposób zbliżony do języka potocznego (najczęściej angielskiego).

Program napisany w **języku wyższego poziomu**, aby mógł być wykonany, musi zostać zanalizowany przez inny program zwany **translatorem** (kompilatorem lub interpreterem) albo programem tłumaczącym. Translator rozkłada program źródłowy (w języku wyższego poziomu) na proste operacje w postaci rozkazów zrozumiałych dla danego mikroprocesora. Różnica pomiędzy kompilatorem a interpreterem polega na tym, że **kompilator** zamienia cały program źródłowy w ciąg rozkazów w postaci dwójkowej, natomiast **interpreter** pobiera po jednej linii programu źródłowego, zamienia ją na kod dwójkowy i zleca wykonanie. Oczywiście program skompilowany wykonuje się dużo szybciej niż program interpretowany.

W ROM-ie większości mikrokomputerów zapisany jest na stałe interpreter języka **Basic**. Dodatkowo dostępne są translatory innych języków wysokiego poziomu — Pascal, Logo, Forth, Fortran, Pilot, Prolog, Ada, C i inne.

WOJCIECH PENCZEK

Słowniczek

adres — ciąg symboli, bitów lub liczba identyfikująca rejestr, komórkę pamięci lub urządzenie

algorytm — jednoznaczny przepis na rozwiązanie zadania

asembler — program komputera używany do tłumaczenia programu wyrażonego w języku adresów symbolicznych na język procesora

bajt — najkrótszy, adresowalny ciąg bitów, traktowany jako niepodzielna całość; składa się z ośmiu bitów (ewentualnie plus bit kontrolny)

bit — cyfra dwójkowa (0 lub 1), jednostka ilości informacji

hardware — sprzęt komputerowy

interfejs — urządzenie wyjścia lub wejścia

kilobajt — kB, 1024 bajty

kod ASCII — obowiązujący w USA i szeroko stosowany w innych krajach znormalizowany kod 8-bitowy (7 bitów informacji i 1 bit parzystości)

kompatybilność — wymienność oprogramowania dla różnych modeli komputerów (zgodność)

megabajt — MB, 1024 x 1024 bajtów

modem — urządzenie umożliwiające transmisję danych przez łącza telekomunikacyjne

procesor — urządzenie w komputerze cyfrowym, które może wykonywać autonomicznie ciąg rozkazów

program — ciąg instrukcji w określonym języku programowania

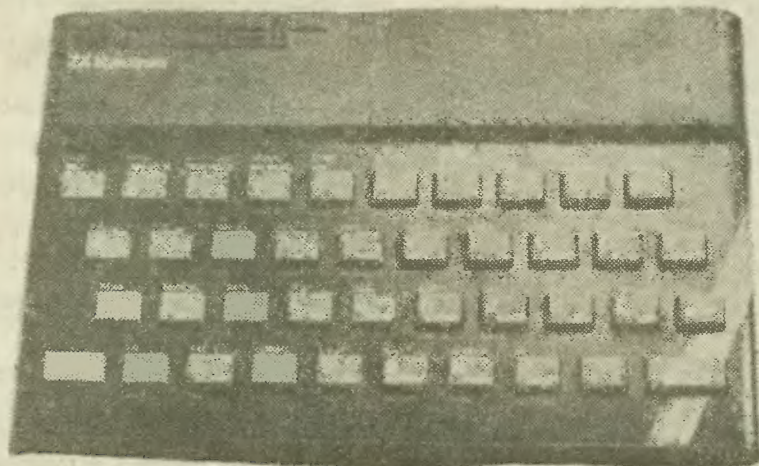
software — oprogramowanie

(r.p.)

Wszystkich zainteresowanych podnoszeniem kultury informatycznej w naszym kraju, osoby posiadające doświadczenie w użytkowaniu mikrokomputerów, wszystkich, którzy chcą z BAJTKIEM współpracować lub choćby wymienić uwagi, prosimy o kontakt z redakcją. Adres do korespondencji:

Redakcja „Sztandaru Młodych"
00-687 Warszawa
ul. Wspólna 61
BAJTEK

Telefony:
29-51-06 (Waldemar Siwiński),
28-52-71 wew. 259 (Roman Poznański).



Budowa ZX Spectrum

„Sercem" ZX Spectrum jest mikroprocesor Z80A lub Z80B firmy Zilog, nowsze wersje popularnego Z80. Jedyną rożnicą między tymi układami jest częstotliwość pracy zegara. Dla Z80 wynosi ona 2,5 MHz, natomiast dla Z80A — 4 MHz (w praktyce 3,5 MHz), a dla Z80B — 6 MHz. Dzięki temu Z80A i Z80B pracują szybciej niż ich poprzednik.

Mikroprocesor zastosowany w ZX Spectrum ma ośmiobitową szynę danych. To znaczy, że wymiana informacji między CPU i pamięcią operacyjną odbywa się w postaci paczek po osiem bitów każda, czyli porcji składających się z kombinacji ośmiu zer i jedynek. Zero jest reprezentowane przez niski stan napięcia 0÷0.8 V, a jedynka przez wysoki stan napięcia 2÷5 V. Dlatego też mikroprocesor Z80A nazwano ośmiobitowym. (Inne procesory ośmiobitowe to 8080 firmy Intel, 6502 firmy Rockwell czy 6800 firmy Motorola).

Z80A ma szesnastobitową szynę adresową. Pozwala ona na wygenerowanie $2^{16}-1 = 65535$ różnych adresów. Wygenerowany adres jednoznacznie wskazuje która z komórek pamięci operacyjnej lub który z układów wejścia/wyjścia ma być w danej chwili zaangażowany w wymianę informacji. W praktyce do oznaczenia układów I/O wykorzystuje się osiem pierwszych bitów adresu, gdyż rzadko zdarza się, aby system wykorzystywał więcej niż 256 układów wejścia/wyjścia. W przypadku ZX Spectrum drugą połowę szyny adresowej wykorzystuje się do generowania adresów służących do odczytu klawiatury.

Inną ważną funkcją procesora jest wytwarzanie sygnałów, wskazujących czy wygenerowany adres został przeznaczony dla pamięci czy też dla układów sprzęgających. Jeżeli wraz z adresem na odpowiednim wyjściu mikroprocesora pojawi się sygnał zwany MREQ (ang. Memory REQuest) to znaczy że w przesłaniu informacji weźmie udział pamięć operacyjna. Natomiast jeżeli z adresem na odpowiednim wyjściu mikroprocesora pojawi się sygnał zwany IORQ (ang. Input Output ReQuest) to znaczy, że zgłasza się jeden z układów wejścia/wyjścia). Należy pamiętać, że szyna adresowa przesyła sygnały tylko z mikroprocesora do układów wejścia/wyjścia lub do pamięci — nigdy odwrotnie. Szyna adresowa jest jednokierunkowa.

Inaczej rzecz się ma z szyną danych. Jest ona dwukierunkowa. Można nią przesłać informacje z i do mikroprocesora. O tym, w którym kierunku ma nastąpić przesłanie decydują sygnały: RD (ang. ReaD) — odczytu z pamięci lub układów wejścia/wyjścia lub WR (ang. WRite) zapisu do pamięci lub przesłania na wyjście. Dokładniej jest to opisane w pracy zbiorowej pt. „Modułowe systemy mikrokomputerowe" wydanej przez WNT w 1984 roku.

ROM i RAM

Przestrzeń adresową ZX Spectrum można podzielić na dwie podstawowe części: ROM (ang. Real Only Memory) i RAM (ang. Random Access Memory). Pamięć ROM zajmuje 16 kB (czyli 16 x 1024 słowa po osiem bitów każde) przestrzeni

adresowej od adresu 0 do 16383. Znajduje się tam system operacyjny zajmujący około 7 kB. Zawiera on między innymi procedury obsługi błędów, klawiatury, urządzeń zewnętrznych, ekranu, edytora itp... Oprócz systemu operacyjnego ROM zawiera interpreter języka BASIC. Zadaniem jego jest „tłumaczenie" programu z języka BASIC na język zrozumiały dla mikroprocesora. Ściślej można powiedzieć, że interpreter wykonuje program napisany w BASIC-u. Programy znajdujące się w obu tych częściach ROM-u napisane są w kodzie maszynowym — assemblerze Z80.

Za interpreterem BASIC-u znajduje się kalkulator. Jego zadaniem jest wykonywanie operacji logicznych i arytmetycznych.

Ostatnie 768 bajtów ROM-u, od adresu 15616 ÷16383 przeznaczone są na generator znaków (ang. Character Generator). W tym obszarze zdefiniowane są kształty wszystkich liter, cyfr i innych znaków, dostępnych z klawiatury. Znaki kodowane są w standardowym kodzie ASCII (ang. American Standard Code for Information Interchange) i zawierają cyfry, znaki specjalne oraz litery angielskie co w konsekwencji powoduje, że bezpośrednio z klawiatury niedostępne są polskie litery takie jak: ą, ć, ę, ś itp.. (chociaż takie znaki można zdefiniować samemu).

ROM służy tylko do odczytu. Jest to tzw. pamięć stała. Nie można nic w niej zmienić. Cechą charakterystyczną systemu operacyjnego umieszczonego w pamięci stałej ZX Spectrum jest to, że po przyłączeniu zasilania następuje automatyczny start programu zawartego w ROM-ie od adresu zero i już po chwili mikrokomputer jest gotowy do pracy.

Zaraz za ostatnim bajtem ROM-u zaczyna się przestrzeń adresowa RAM-u. W wersji podstawowej zajmuje ona 16 kB. Właściciel ZX Spectrum 16 kB może samodzielnie rozszerzyć pamięć do 48 kB gdyż na płytce „drukowanej" znajdującej się wewnątrz mikrokomputera jest zarezerwowane miejsce na dodatkowe układy scalone pamięci. Produkcja wersji 16 kB została już dawno przez firmę Sinclair zaniechana.

Najważniejszą cechą pamięci RAM jest możliwość zmiany zawartości jej komórek. Każdy napisany przez użytkownika program lub wczytany z pamięci zewnętrznej (np. z magnetofonu) oraz dane, umieszczone są w tej części pamięci. Jeżeli od mikrokomputera odłączy się zasilanie, cała zawartość pamięci RAM ulegnie bezpowrotnemu zniszczeniu, odwrotnie niż pamięci ROM.

Mimo że cała zawartość pamięci RAM może być zmieniana, to istnieją pewne obszary (zmienne systemowe, mapa microdriva — jeśli jest podłączony, UDG itd...), w których po załączeniu zasilania system operacyjny umieszcza pewne dane, potrzebne do jego pracy. Niektóre z tych wartości możemy zmieniać sami, co niekiedy bywa bardzo użyteczne. Jednak należy robić to rozważnie po wcześniejszym zapoznaniu się z systemem operacyjnym ZX Spectrum.

ULA

Bardzo ważnym układem omawianego mikrokomputera jest ULA (ang. Uncommited Logic Array). Można porównać ją do dużego węzła łączności. ULA jest połączona z mikroprocesorem jako układ wejścia/wyjścia o adresie 254 (tylko zerowy bit pierwszego bajtu adresu równy zero). ULA jest także połączona z szyną danych ZX Spectrum np. do jednej linii szyny danych podłączony jest głośnik. Można się o tym przekonać realizując poniższy program

```
10 OUT 254, 16:OUT 254, 0:GOTO 10
```

W wyniku tego wykonania będzie słyszalne niskie brzęczenie dobiegające z głośnika wewnętrznego Spectrum. Zatrzymanie programu nastąpi po jednoczesnym przyciśnięciu klawiszy CAPS SHIFT I BREAK. W trakcie wykonywania programu zmieni się także kolor granicznej części ekranu telewizora tzw. BORDER, na czarny. Wynika z tego, że ULA zajmuje się kontrolą koloru brzegu ekranu.

Wyczytajmy następujące programy:

a)
```
10 FOR n = Ø TO7
2Ø OUT 254, n:PAUSE 1ØØ
3Ø NEXT n
```

b)
```
1Ø FOR n = ØTO 7
2Ø BORDER n:PAUSE 1ØØ
3Ø NEXT n
```

W czasie wykonywania się programów będzie zamieniał się kolor brzegu ekranu co około 2 sek. Efekt działania obu programów jest identyczny. Jeżeli po wykonaniu programu pierwszego a) naciśnie się ENTER, to wowczas kolor granicznej części ekranu powróci do wcześniej zdefiniowanego, tzn. do koloru określonego przed uruchomieniem programu a). Natomiast po wykonaniu drugiego programu b) kolor BORDER-u pozostanie taki jak został określony kodem, koloru ostatnio wykonanej instrukcji BORDER. Stąd wniosek, że zmiany prowadzone z pomocą OUT 254, n, są tylko zmianami bieżącymi i nie zostawiają one w pamięci komputera żadnych śladów.

Do tego samego wyprowadzenia linii szyny danych, co głośnik podłączony jest pakiet dopasowujący magnetofon. Jest to układ umożliwiający bezpośrednie połączenie magnetofonu z mikrokomputerem.

Podsumowując: bity 0, 1, 2 szyny danych sterują przez ULA kolorem granicznej części ekranu. Bit 3 kontroluje wyjście z komputera na magnetofon a bit 4 kontroluje głośnik.

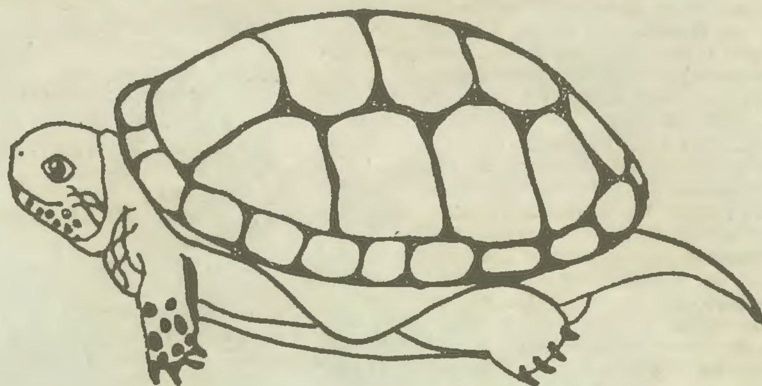
Do tej pory omówiono ULA pracującą jako urządzenie wyjściowe. Może ona także pracować jako urządzenie wejściowe. ULA steruje przesłaniem danych z gniazda EAR (czyli z wejścia magnetofonu) do pamięci mikrokomputera. Jest ona także urządzeniem pośredniczącym między mikroprocesorem, pamięcią a klawiaturą.

ULA wczytuje także informacje z pamięci obrazu, i na ich podstawie generuje kolorowy obraz telewizyjny, sterując pracą kodera PAL-owskiego i modulatora UHF. Odbywa się to z częstotliwością 50 razy na sekundę. Dlatego ma się wrażenie, iż obraz telewizyjny jest stały, nie migający — analogiczna sytuacja jest z obrazem wyświetlanym np. z projektora filmowego.

Wszystkie omówione wyżej — prócz ULA — elementy mikrokomputera są dostępne na rynku nawet w krajach socjalistycznych. ULA natomiast jest specjalizowanym układem wykonanym na zamówienie SINCLAIR-a. Tego typu układy bywają niekiedy bardziej skomplikowane niż mikroprocesory. Zdecydowana większość komputerów domowych ma choć jeden wyspecjalizowany układ podobny do ULA. Jest to więc jeden z najbardziej istotnych elementów mikrokomputera.

KRZYSZTOF KURYŁOWICZ
DARIUSZ MADEJ

LOGO LOGO LOGO LOGO



> *„... Komputery aż do dwudziestej generacji włącznie odznaczały się „owadzim" zachowaniem: nie mogły kwestionować, a tym bardziej — przekształcać swoich programów. Programista „impregnował" swoją maszynę wiedzą jak Ewolucja „impregnuje" owada — instynktem (...). Edukacja komputera osiemdziesiątej generacji była już daleko bardziej podobna do w y c h o w a n i a dziecka aniżeli do klasycznego programowania maszyny cyfrowej".*
>
> *(Stanisław Lem „GOLEM XIV")*

LOGO uważany jest za język najbardziej nadający się do pierwszego kontaktu z komputerem. Nie oznacza to jednak, że LOGO jest czymś w rodzaju „niby języka", czymś gorszym niż BASIC. LOGO zostało tak zbudowane, by już początkujący mógł posługiwać się środkami i metodami typowymi dla „dorosłej" informatyki, oglądając przy tym wyniki swoich prób natychmiast na ekranie. LOGO nie jest jednak wyłącznie udziwnionym sposobem rysowania obrazków. Z powodzeniem można w LOGO tworzyć poważne programy z myślą o „prawdziwych" zastosowaniach. Przykłady takich programów opublikujemy wkrótce.

Nie istnieje dotychczas uzgodniona, standardowa, polska wersja LOGO. Dlatego, komentując poszczególne komendy, przy niektórych podajemy proponowane polskie odpowiedniki. stosowane w niektórych przeróbkach.

Program w LOGO składa się ze **słów**

słowo

Słowem może być dowolny ciąg znaków nie zawierający odstępów, nawiasów, cudzysłowów i znaków działań arytmetycznych. Słowo może zawierać któryś z tych znaków tzw. separatorów, lecz wówczas musimy poprzedzić ten znak znakiem **backshlash.**

separatory

element słowa

Znaki składające się na słowo nazywamy jego **elementami.**

słowo puste

Słowo nie zawierające żadnego znaku nazywa się słowem **pustym.** Oznaczamy je " .

Słowa w LOGO mogą być nazwami **procedur, zmiennych** lub elementami wyrażeń algebraicznych, których wartości są **parametrami** procedur.

linia

Słowa w programie grupowane są w **linie,** przy czym grupowanie ma najczęściej charakter czysto porządkowy.

lista

Oprócz słów program może zawierać **listy,** w skład których mogą wchodzić słowa lub inne listy. Obszerniej o budowie słów i list powiemy dalej.

procedura
parametr
obiekt

Słowo użyte w programie musi być nazwą **procedury,** jej **parametru** lub jej **obiektem.** Parametry i obiekty procedur występują zawsze za ich nazwami.

nazwa
słowo kluczowe
procedura podstawowa
definiowanie procedury

Słowo jest **nazwą procedury,** jeśli jest **słowem kluczowym,** czyli nazwą **procedury podstawowej** (primitive) i jako takie jest zdefiniowane w ramach samego programu tłumaczącego LOGO lub jeśli uprzednio została **zdefiniowana** procedura o tej nazwie.

Słowo użyte w programie bezpośrednio, tzn. bez cudzysłowu lub dwukropka i nie w ramach listy zawsze jest traktowane jak nazwa **procedury,** którą należy wykonać.

Procedurę definiuje się przy pomocy słowa kluczowego TO (TO DRZEWO). Definicja musi się kończyć słowem END, napisanym w osobnej linii. Wprowadzone uprzednio definicje można zmienić korzystając z **trybu edycji,** który uzyskuje się przy pomocy słowa kluczowego EDIT (np. EDIT "DRZEWO).

EDIT

Podanie EDIT bez wskazania, o którą procedurę chodzi powoduje zawsze przywołanie ostatnio edytowanej procedury. Można również przywołać pusty edytor podając jako parametr wejściowy EDIT listę pustą (EDIT [ ]) i wykorzystać go do definiowania nowej procedury. można też wyedytować kilka procedur równocześnie podając jako parametr EDIT całą listę nazw (np. EDIT [DRZEWO DOM PŁOT]).

W trybie edycji można przesuwać kursor strzałkami, wstawiać nowe znaki i usuwać je przy pomocy DELETE. Strzałki w EXTEND MODE przesuwają kursor na początek i koniec linii (lewo-prawo) oraz na początek i koniec tekstu (góra-dół). EXTEND MODE Y usuwa wszystkie znaki danej linii leżące na prawo od kursora i zapamiętuje je, a EXTEND MODE R — wstawia je w nowe miejsce.

Jeśli edytowany tekst nie mieści się na ekranie, wówczas można jego przesuwanie się zatrzymać przy pomocy SYMBOL SHIFT Y. EXTEND MODE P powoduje wówczas powrót do poprzedniej strony, a EXTEND MODE N - przejście do następnej.

wyjście z edycji

Tryb edycji można opuścić przy pomocy zlecenia EXTEND MODE C i wówczas wszystkie edytowane procedury otrzymają nowe definicje lub przy pomocy BREAK i wówczas wszystkie wprowadzone podczas edycji zmiany zostają zapomniane. Przy pomocy słowa kluczowego ENDS [lista nazw] można również wyedytować zmienne wraz z ich wartościami, po czym zmienić ich nazwy i wartości.

ENDS

parametry procedury

Zdefiniowana procedura może wymagać podania **parametrów.** W programie parametry podajemy bezpośrednio za jej nazwą, w definicji zaś trzeba je zadeklarować w pierwszej linii, podając ich nazwy po dwukropku (np. TO DRZEWO :Wysokość :liczba gałęzi :grubość pnia). Parametrów może być dowolnie dużo. Parametr jest **zmienną lokalną** w ramach swej procedury i procedur wywoływanych przez nią.

procedury funkcje OUTPUT OP

Procedura może po wykonaniu przyjmować pewne wartości wyjściowe i zachowuje się wówczas jak **funkcja.** Jeśli w treści procedury znajduje się słowo OUTPUT lub krócej OP, wówczas jej realizacja zostaje zakończona, a wartość wyrażenia następującego po tym słowie zostaje przekazana do procedury nadrzędnej (uwaga: każda procedura, niezależnie od sposobu kończenia realizacji. musi kończyć się słowem END).

STOP

TOPLEVEL

Realizacja procedury przed osiągnięciem END zostaje przerwana także w razie natrafienia za zlecenia STOP. W takim przypadku LOGO powraca do procedury bezpośrednio nadrzędnej (więc do tej, z której aktualnie realizowana procedura była wywoływana). Natomiast zlecenie TOPLEVEL zawsze przerywa realizację programu i powoduje powrót do najwyższego poziomu wywoływania, a więc w najczęstszym wypadku posługiwania się programem interpretującym LOGO — po prostu do stanu oczekiwania na zlecenie z klawiatury. TOPLEVEL jest więc odpowiednikiem zlecenia STOP w BASICU, podczas gdy STOP działaniem przypomina RETURN. Realizację każdego programu można również przerwać bezpośrednio z klawiatury przy pomocy BREAK, co wywołuje komunikat „STOPPED!!!".

Po zdefiniowaniu procedury uzyskujemy **komunikat,** np. „DRZEWO DEFINED" (lub „DRZEWO zdefiniowano"). Do wywołania procedury wystarcza teraz użycie jej nazwy w programie. Jeśli procedura wymaga podania parametrów, podajemy je zaraz za nazwą. Gdy zabraknie parametrów LOGO przerywa realizację programu komunikatem „Not enough inputs to DRZEWO" lub „za mało danych dla DRZEWO", gdy natomiast podano ich za dużo komunikatem „YOU don't say what to do with" lub „nie wiem, co zrobić z...".

Próba nadania nowo zdefiniowanej procedurze takiej samej nazwy, jaką ma procedura zdefiniowana już uprzednio kończy się komunikatem **„DRZEWO is already defined"** lub „DRZEWO jest już zdefiniowane". Trzeba w takim przypadku zmienić nazwę nowej procedury lub usunąć starą. Próba użycia jako nazwy słowa kluczowego kończy się komunikatem **„FORWARD is used by LOGO"** lub „NAPRZÓD jest zastrzeżone dla LOGO".

POTS

Listę już zdefiniowanych procedur można obejrzeć (po uprzednim przejściu na tekstowy tryb organizacji ekranu!) dzięki słowom klubowym POTS (Print Out the TitleS), jeśl

POPS

PO
POALL

interesują nas tylko nazwy procedur oraz POPS (Print Out Procesure S), jeśli interesują nas ich pełne definicje. Możemy również obejrzeć treść jednej, wybranej procedury — PO nazwa, np. PO "DRZEWO oraz listę zmiennych — PONS (Print Out the NameS) wraz z ich wartościami. Natomiast POALL (Print Out ALL) powoduje wyświetlanie wszystkich zdefiniowanych procedur oraz zmiennych.

CONTENTS

Całą zawartość pamięci można również przejrzeć przy pomocy zlecenia .CONTENTS (zwartość), które jednak wymaga dużego obszaru wolnej pamięci. .CONTENTS (pamiętaj o kropce na początku!) podaje nie tylko zdefiniowane procedury i zmienne, ale również większość tekstów wprowadzonych z klawiatury i rozkazy wykonywane bezpośrednio.

PRIMITIVES

Listę słów kluczowych można natomiast przypomnieć sobie przy pomocy zlecenia .PRIMITIVES.

ERASE
ER ERALL
ERNS ERPS
ERN

Usunąć procedurę z pamięci można przy pomocy zlecenia ERASE lub krótko ER, np. ER "DRZEWO. Zlecenia ERALL ERNS, ERPS mają znaczenie analogiczne do odpowiednich konstrukcji z PO. Do usuwania zmiennych używamy zlecenia ERN nazwa, np. ERN "liczbagałęzi.

DEFINEDP
PRIMITIVEP

Jeśli mamy wątpliwości czy dane słowo jest już użyte jako nazwa lub słowo kluczowe możemy również skorzystać ze zleceń DEFINEDP "DRZEWO lub PRIMITIVEP "DRZEWO. Zlecenia te są pomyślane jako środek używany w ramach innych procedur, tak więc po wykonaniu przyjmują (jak OUTPUT) wartość TRUE lub FALSE. Jeśli chcemy skorzystać z nich bezpośrednio, musimy powiedzieć LOGO, co ma uczynić z wynikiem tej operacji pisząc np. PRINT DEFINEDP "COŚ.

DEFINE TEXT

COPYDEF

Innymi środkami umożliwiającymi tworzenie programów samomodyfikujących się, a więc definiowanie nowych procedur w trakcie realizacji programu, są zlecenia DEFINE (np. DEFINE "SUMA[[:X:Y][PRINT:X+:Y]]), TEXT (np. PRINT TEXT "SUMA daje [:X:Y][PRINT :X+:Y oraz COPYDEF, która tworzy nową procedurę o nowej nazwie i treści identycznej z treścią wskazanej istniejącej procedury, np. COPYDEF "DODAC "SUMA utworzy nową procedurę.

```
TO DODAC  :X:Y
PRINT     :X+:Y
END
```

Przykład ten wskazuje, że mimo iż w zleceniach DEFINE i TEXT używamy nieco innego sposobu zapisu procedury: lista parametrów procedury i jej kolejne linie przedstawione są jako lista list, nie używa się natomiast END, to jednak w pamięci zdefiniowana procedura ma zawsze taką samą postać.

rekurencja

Procedury w LOGO mają wywoływać same siebie, możliwa jest więc tzw. **rekurencja.** Jeśli wywoływanie nie wiąże się z przekazywaniem parametrów do procedury nadrzędnej, wówczas rekurencja może być wykonywana dowolną ilość razy (np. TO ZOLWLATAWKOLKO FD 3 RT 3 ZOLWLATAWKOLKO END). Uwaga! lokalność parametru procedury jest przestrzegana bardzo starannie i w wypadku rekurencji mimo tej samej nazwy parametr na każdym stopniu wywołania zachowuje własną wartość, np. :TO SZEREG:LICZBA IF :LICZBA 100 SZEREG :LICZBA 3 PR :LICZBA END daje w wy-

lokalność
parametru

niku wywołania SZEREG 5 następujący wynik: 135 45 15 5 Przeanalizuj dokładnie ten przykład!

zmienne są globalne
MAKE

W LOGO wszystkie zmienne poza parametrami procedur są **globalne.** Zmienne tworzy się i nadaje się im wartość przy pomocy zlecenia MAKE, np. MAKE A 1 tworzy zmienną o nazwie A i nadaje jej wartość 1. Wartością zmiennej może być liczba lub lista, która może być z kolei oczywiście listą liczb, list zmiennych lub słów, albo wręcz wszystkiego po trochu równocześnie. Od użytkownika nie wymaga sie uprzedniego zadeklarowania charakteru zmiennej.

THING

Posługując się zmienną poprzedzamy jej nazwę dwukropkiem, jeśli chcemy posłużyć się jej wartością — :A znaczy wartość zmiennej A. Zamiast dwukropka możemy użyć słowa kluczowego THING. Gdy interesuje nas nie — wartość, a nazwa zmiennej (podobnie w wypadku procedur — poprzedzamy ją znakiem ". Samo A np. A nie poprzedzone żadnym znakiem zostałoby przez LOGO zrozumiane, jako wywołanie procedur o nazwie A.

separatory

= + − ( ) "

Słowa w LOGO oddzielone muszą być **separatorem.** Najprostszym separatorem jest odstęp — pamiętaj. że np. nazwa procedury i jej parametr w wypadku jej wywołania to dwa oddzielne słowa i DRZEWO 5 znaczy coś zupełnie innego niż DRZEWO5 (to ostatnie na ogół nic nie znaczy). LOGO nie wie również, jak zrobić np. RT45 bez uprzedniego zdefiniowania, gdyż słowem kluczowym RT należy posługiwać się tak: RT 45.

listy

Separatorami są też nawiasy zwykłe używane przy zapisie wyrażeń arytmetycznych oraz instrukcji **zachlannych** (patrz dalej) oraz nawiasy kwadratowe oznaczające listy i znaki działań arytmetycznych. Oznacza listę złożoną z obiektów zawartych wewnątrz nawiasu.

−5 i − 5

Separatory nie muszą być oddzielane od sąsiednich słów odstępami, zalecane jest jednak oddzielenie odstępami liczb i oznaczeń arytmetycznych ze względu na możliwość nieporozumienia: —5 znaczy liczba minus pięć, natomiast — 5 znaczy operator odejmowania i liczba 5 jako odjemna. Różnica polega na tym, że np. 7 — 5 jest jedną liczbą równą 2, natomiast 7—5 to para dwóch liczb, efektem pomyłki jest więc zazwyczaj komunikat, że podano za mało parametrów bądź LOGO nie wie, co robić z niepotrzebną mu dodatkową liczbą.

instrukcje zachłanne

LIST
WORD

SENTENCE

Niektóre instrukcje są **zachlanne:** umieszczane w nawiasie okrągłym wraz z grupą obiektów obejmują swym działaniem całą tę grupę, np. (PRINT "Ala "ma" kota) daje w wyniku Ala ma kota. Podobnie (LIST "Ala "ma "kota) jest równoważne [Ala ma kota], a WORD ("Ala "ba"ma) znaczy "Alabama. (SENTENCE "Ala "ma "kota) również jest — podobne jak LIST — równoważne [Ala ma kota], ale wynikiem operacji SENTENCE zawsze jest lista złożona ze słów, a nie z innych list. Ewentualnie listy składowe SENTENCE traktuje po prostu jako grupy słów

W. M

ATIC - ATAC

Wyobraźcie sobie, że gdzieś za górami i lasami stoi potężne i posępne zamczysko, kryjące najbardziej niewiarygodne tajemnice. Stara legenda powiada, że upstrzone blankami mury kryją w sobie między innymi rozwiązanie zagadki wiecznego życia, ukrytej za wielkimi karmazynowymi drzwiami. Otworzyć je zdola jedynie ten śmiałek, który zdobędzie trzy części wielkiego klucza. Nikomu nigdy nie udało się zdobyć choćby jednej z nich, lecz wszyscy wiedzą o nich i wierzą w ich istnienie.

Ci, którzy pragną spróbować szczęścia, przybywają w tę okolicę ze wszystkich stron świata, nikt jednak nie widział, by któryś z tych młodych ludzi powrócił z wyprawy. Gdy tylko kolejny śmiałek przekroczy wrota zamczyska, nieznane siły przenoszą go przed same tajemnicze wielkie drzwi. Teraz nie ma już odwrotu: kto nie odnajdzie wszystkich części wielkiego klucza, ten musi zginąć. Nim jednak rozpocznie poszukiwania, duch zamku pozwoli mu wybrać jedną z trzech ról: rycerza, czarownika lub chłopa, dając w ten sposób zadziwiającą możliwość przechodzenia przez ściany z półkami bibliotecznymi, z baryłkami wina lub ze starymi wahadłowymi zegarami. Najdziwniejsze jest to, że przechodząc zegar lub beczkę trafiasz nieraz w odległy punkt zamku, choć wydaje ci się, że uczyniłeś jeden krok. Przechodząc przez konkretny obiekt, trafiasz jednak zawsze w to samo miejsce.

Niestety, raz dokonawszy wyboru roli możesz przechodzić tylko przez jeden rodzaj obiektów — i choćbyś żałował potem swej decyzji, nie możesz już jej zmienić.

Legenda głosi, że zamczysko ma cztery piętra komnat, a pod nimi rozległe posępne piwnice, w których natrafić można na białe kości poprzedników lub straszydła, gotowe skorzystać z każdej chwili nieuwagi, by wyssać trochę twej energii życiowej.

Aby znaleźć wszystkie części wielkiego klucza musisz przeszukać wszystkie komnaty i przejścia zamczyska, co nie jest łatwe, gdyż drzwi do wielu z nich są pozamykane. Musisz więc najpierw odnaleźć zwykłe klucze, które dawni mieszkańcy uchodząc z zamku pogubili w różnych miejscach. Klucze te ozdobione są różnokolorowymi uchwytami i jak głosi legenda, klucz o danym kolorze otwiera stare drzwi w tym samym kolorze, nawet jeśli są one zabite deskami, pokryte pajęczynami i zamek w nich od lat jest zardzewiały. Jedynym wyjątkiem są drzwi białe, które na ogół są otwarte, często jednak w tajemniczy i kapryśny sposób same się zatrzaskują i nie można ich wówczas otworzyć. Trzeba po prostu poczekać aż otworzą się ponownie. Żaden pojedynczy klucz nie otworzy WIELKICH TAJEMNICZYCH WRÓT, jedynych, przez które można opuścić zamczysko unosząc ze sobą jego skarby i wielką tajemnicę.

Czekające cię zadanie nie jest łatwe także dlatego, że komnaty i przejścia zamczyska od dawien dawna zamieszkują różne, rzadko już w naszych czasach spotykane potwory: demony, drakule, frankensteiny i ich kumple. Natrafiwszy na takiego przyjemniaczka zawsze masz dwie możliwości: uciec lub zabić go (jeśli masz przy sobie broń), co jednak nie zawsze jest możliwe, gdyż musisz pamiętać, że jesteś w stanie udźwignąć i nosić ze sobą jedynie trzy przedmioty. Jeśli więc masz już przy sobie np. trzy klucze, nie możesz nieść broni. Często trzeba wybierać, a wybór nie jest łatwy.

Poszukując trzech części wielkiego klucza nie możesz marnować czasu, gdyż zginiesz z głodu, zanim cokolwiek znajdziesz. Na szczęście przemierzając kolejne pomieszczenia masz szanse znaleźć w nich różne smaczne potrawy, których spożycie natychmiast doda ci nowych sił. Musisz korzystać z tych zasobów bardzo racjonalnie, gdyż pełne rozwiązanie twego zadania wymagać może od ciebie utrzymania się przy życiu przez długi czas, a potraw w zamku nie przybywa.

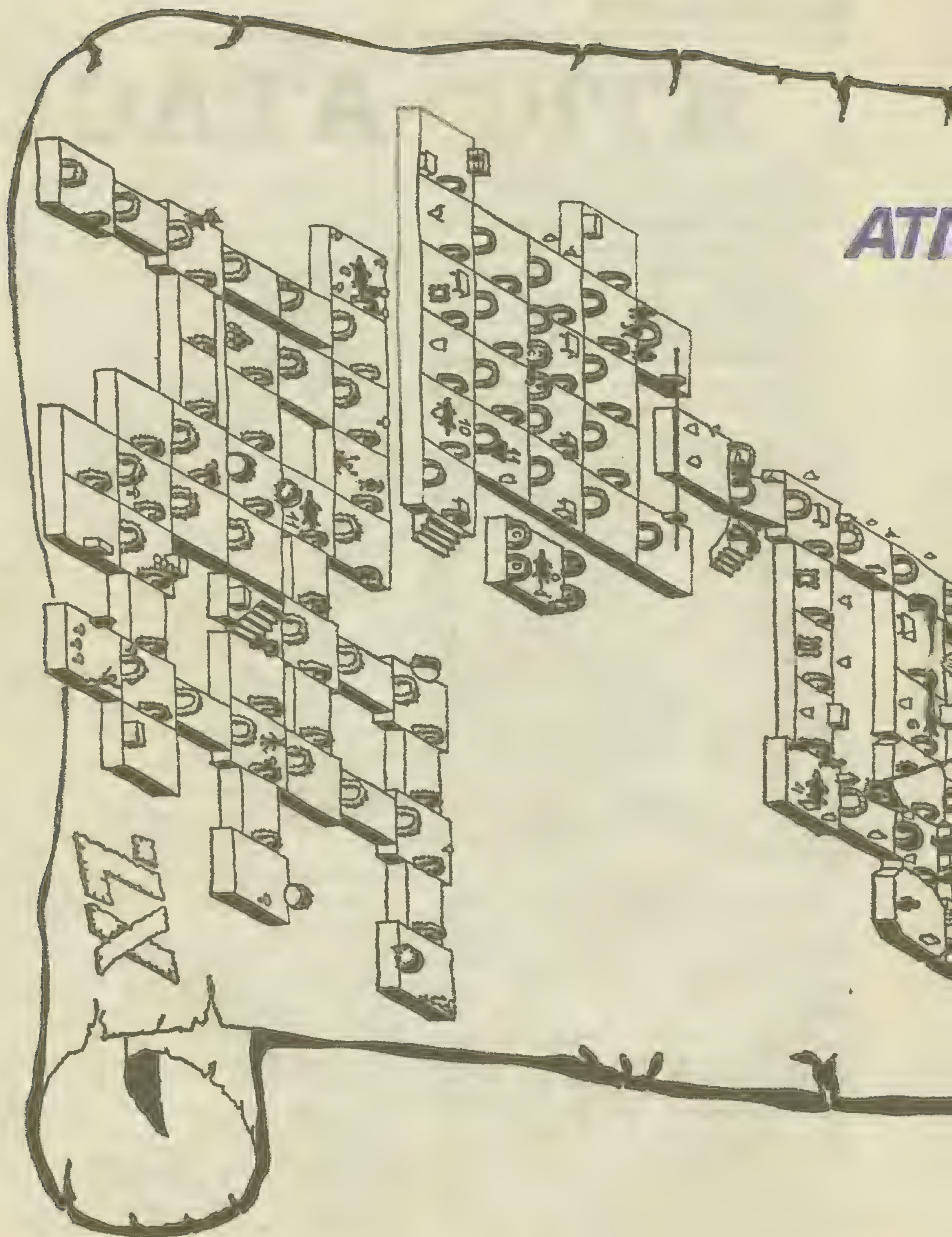
Może to właśnie ty odkryjesz kiedyś najgłębszy sens niezrozumiałego dziś porzekadła, od pokoleń powtarzanego przez najstarszych okolicznych mieszkańców: „Symbol shift, by brać i break space, by zatrzymać się i zastanowić". Nikt nie potrafi dziś pojąć tych słów, odkąd zaginął stary pergamin z planem zamku i jego tajemniczymi przejściami.

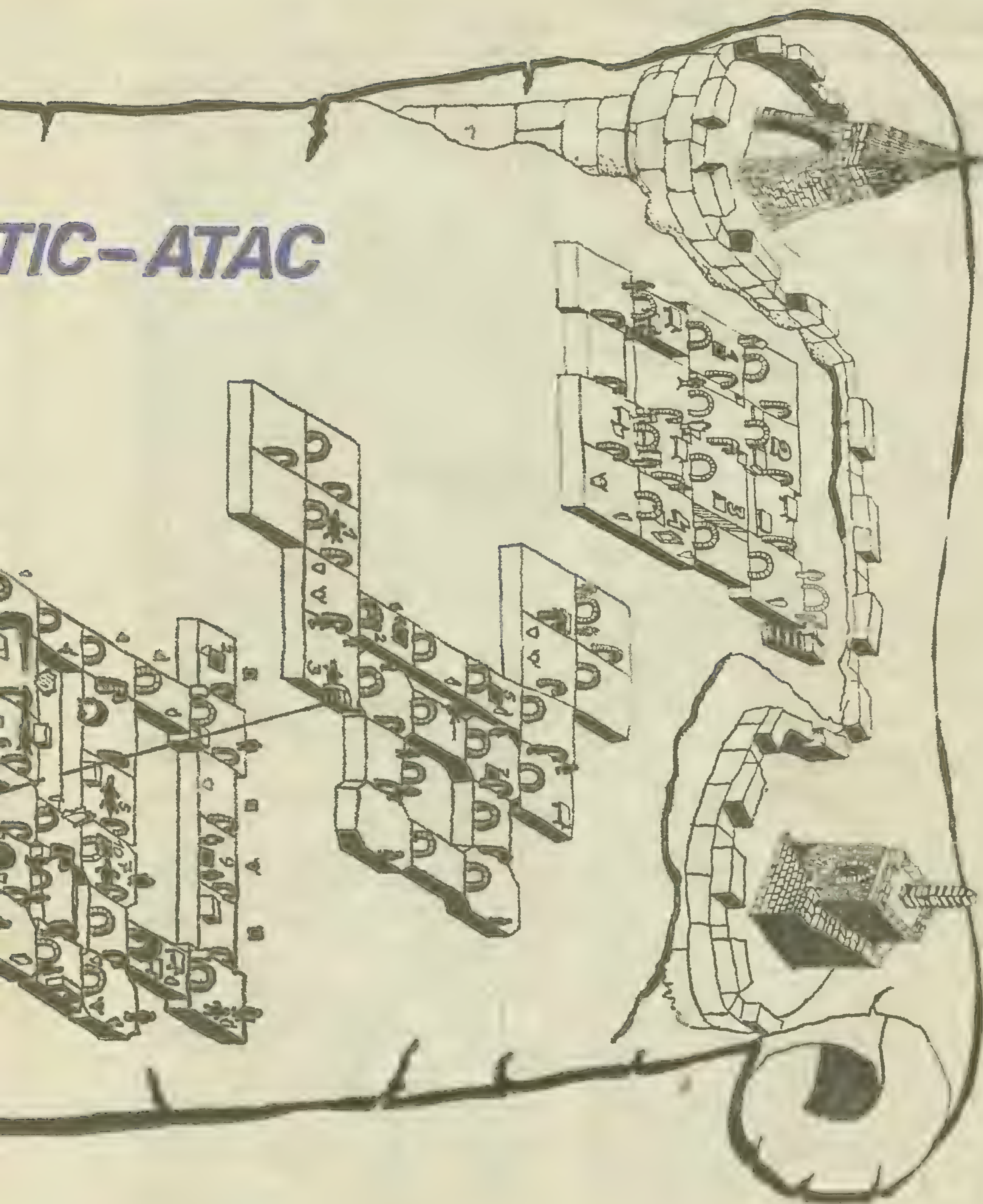
A czy ty chciałbyś poznać jego treść? Już na następnej stronie mamy dla ciebie tę niespodziankę. Czy pomoże ona ci odnaleźć tajemnicę? Napisz do nas, co osiągnąłeś!

*

ATIC-ATAC był na rynku gier komputerowych przebojem 1983 r. Wersja na Spectrum, rozprowadzana przez firmę ULTIMATE PLAY THE GAME liczy ok. 30 kB i jest bardzo rozpowszechniona w naszym kraju. Gra jest szybka, ma bogatą grafikę, urozmaiconą konstrukcję i szybko urzeka każdego, kto może spędzić kilka godzin sam na sam z komputerem.

W.M.


ATI
XL


TIC–ATAC

CZEGO UCZYĆ?

Począwszy od roku szkolnego 1986/87 w liceach ogólnokształcących i wielu rodzajach techników wprowadzony będzie nowy przedmiot uzupełniający — elementy informatyki. Będzie to jednoroczny kurs prowadzony w szkołach dysponujących odpowiednim wyposażeniem. W innych placówkach uczniom pozostaną do wyboru uzupełniające zajęcia m.in. z prawa lub podstaw ekonomii.

Ten pierwszy krok wzbudził niepokój wśród wielu obserwatorów przekonanych, że jeśli polska oświata za coś się bierze, musi to być sknocone. Przekonanie to wyrażano publicznie, tak więc jako uczestnik prac zespołu powołanego przez Polskie Towarzystwo Informatyczne, któremu Instytut Programów Szkolnych zlecił opracowanie programu nowego przedmiotu, czuję się w obowiązku ukazać przesłanki, którymi kierowaliśmy się i wnioski do których doszliśmy — w nadziei, że być może nadesłane uwagi pomogą nam usunąć wiele niedoskonałości naszego opracowania, które — czego jesteśmy świadomi — powstawało w zbyt wielkim pośpiechu, choć staraliśmy się utrzymać możliwie poprawny tok prac, w których wzięło udział kilkudziesięciu uczestników.

Podstawowe założenia

Pierwszy etap zbierania propozycji doprowadził do przyjęcia kilku założeń:

- Celem „elementów informatyki" nie powinno być nauczanie o komputerze jako obiekcie technicznym — a więc o jego budowie, działaniu, historii czy też projektowaniu i obsłudze na poziomie profesjonalnym. Tak więc „elementy informatyki" tylko pośrednio spełniać będą rolę politechnizacyjną rozumianą tradycyjnie jako kształcenie „złotych rączek".
- Nie należy również zamieniać nowego przedmiotu w kurs programowania w jakimś konkretnym języku np. Basicu, Logo czy Fortranie, gdyż przydatność takiego kursu byłaby ograniczona, a w dodatku dla znacznej części uczniów zdobyte umiejętności byłyby mało przydatne.
- Zasadniczym natomiast celem zajęć powinno być nauczanie metod rozwiązywania prostych problemów z życia codziennego przy pomocy mikrokomputera, przy czym konkretne urządzenie i język programowania powinny być traktowane jedynie jako środki niezbędne do realizacji tak określonego głównego celu.

Naszym zamierzeniem było więc danie ogółowi młodzieży z liceów ogólnokształcących czegoś w rodzaju „komputerowego prawa jazdy", jednak bez kursu obsługi silnika.

Na pierwszym etapie prac przyjęto też podstawowe założenia co do sprzętu niezbędnego do realizacji programu i języka wykorzystywanego do tego celu. Stanęliśmy przed odwiecznym dylematem: przepaścią między tym, co w szkołach jest, a tym, co chciałoby się mieć. Zimny realizm skłonił nas do wybrania... tego co chciałoby się mieć. Takie założenie będzie bowiem pewną ochroną przed próbami realizowania tego programu od razu w zbyt dużej liczbie nie przygotowanych do tego placówek, pozwoli rozpocząć od etapu prób, a równocześnie ochroni uczniów przed kłopotami z nie dość dojrzałym sprzętem. Szybki spadek cen każe zakładać, że krajowy przemysł będzie w stanie (jeśli będzie miał ku temu dostateczną zachętę) stworzyć w ciągu 2—3 lat tani i dobry mikrokomputer.

Mając świadomość, że obecnie w szkołach jedynie ZX Spectrum 16K i MERITUM można znaleźć w liczbie niezbędnej do rozpoczęcia zajęć przyjęliśmy założenie, iż warunkiem rozpoczęcia kursu jest posiadanie przez szkołę mikrokomputerów dysponujących co najmniej 48K RAM z polską, odporną klawiaturą z pełną grafiką, o konstrukcji umożliwiającej samodzielne ćwiczenia młodzieży, i z pożądaną co najmniej jedną w każdej szkole szybką pamięcią masową oraz drukarką. Takich urządzeń potrzeba tyle, by podczas zajęć w grupach przy jednej klawiaturze siedziało nie więcej niż 3 uczniów, a po lekcjach możliwe były samodzielne ćwiczenia.

Podobne założenia przyjęliśmy przy wyborze języka: zdecydowaliśmy się ani razu w tekście nie wymieniać Logo z nazwy koncentrując się na omawianiu zagadnień możliwych do zrealizowania przy pomocy dowolnego nowoczesnego języka, jednak dobór tematów dostosowany jest do Logo i właśnie w tym języku wydaje się najbardziej naturalny. Program nasz można więc realizować i bez odwoływania się do Logo, najłatwiej jednak będzie się posłużyć tym właśnie środkiem. Logo jest już zresztą dziś najbardziej rozpowszechnionym i najłatwiej dostępnym z wchodzących w grę języków.

Drugi etap prac doprowadził do wyodrębnienia umiejętności, jakie uczeń powinien zdobyć w trakcie kursu, umiejętności niezbędnych do posłużenia się komputerem w przyszłości lub najczęściej wykorzystywanych. Z wielu propozycji wybraliśmy, poza oczywistym wprowadzeniem typu — jak to włączyć, jak uruchomić program i jak go przerwać, trzy główne grupy problemów: tworzenie grafiki komputerowej, działania na strukturach danych oraz redagowanie tekstów.

Taki dobór problemów może wydać się dziwny: oto posługiwania się maszyną, zwaną do niedawna maszyną matematyczną, chcemy uczyć na przykładach spoza tradycyjnie pojmowanej matematyki... Jest to wynikiem dwóch założeń: nie należy mylić elementów informatyki z zastosowaniem komputera do nauczania poszczególnych przedmiotów np. matematyki, a jeśli już jest to nieuniknione, to lepiej posłużyć się przykładami z przedmiotów, których nauczyciele mniej chętnie sięgać będą po to narzędzie.

Po drugie, dążyliśmy do wyboru przykładów które pozwolą uczniowi możliwie najszybciej ujrzeć na ekranie przekonywający, utrwalający się w pamięci efekt własnego działania — a tymczasem docenić piękno i wygodę komputerowych wyliczeń potrafi na ogół jedynie ten, kto poznał smak liczenia „na piechotę"...

Program nauczania

Program składa się więc z 8 „rozdziałów", z których na pierwsze trzy przewidujemy po 5—8 proc. czasu nauczania, a na pozostałe pięć m.in. wprowadzanie komend i tekstów, kasowanie — po ok. 15 proc. czasu. Oto one:

1. Obsługa mikrokomputera, pierwsze kroki m.in. wprowadzanie komend i tekstów, kasowanie pomyłek, możliwe nieprawidłowości w pracy maszyny i postępowanie w razie ich wystąpienia. W tej fazie chodzi o umożliwienie uczniom posługiwania się sprzętem bez ryzyka uszkodzenia maszyny i zapisanych w pamięci zbiorów.

2. Praktyczne zastosowanie mikrokomputera. Ucząc metody strukturalnej budowy oprogramowania nie możemy w inny sposób konstruować samego kursu. Na tym etapie więc na przykładzie gotowych, szeroko stosowanych programów pokazujemy typowe zastosowania informatyki ucząc zasad posługiwania się nimi i analizując ich działanie: określając jakie problemy można przy ich pomocy rozwiązywać, badając czy dają się one podzielić na mniejsze fragmenty oraz zastanawiając się, w jaki sposób są one połączone. Uczeń powinien zdobyć umiejętność posługiwania się oprogramowaniem w trybie dialogu z komputerem. Celem jest pokazanie, że mikrokomputer nie jest tylko przyrządem do gier i obliczeń, że może mieć różnorodne zastosowania. Pokazywane programy powinny umożliwić współpracę przez prosty dialog, nie mogą się sprowadzać do biernie obserwowanych pokazów. Nie jest konieczne, by wszyscy uczniowie opanowali posługiwanie się wszystkimi pokazywanymi na zajęciach programami, ale wszyscy powinni spróbować współpracy z jakimś gotowym programem.



3. Rysowanie. Do rozpoczęcia od grafiki skłania oczywiście to, że jest ona najbardziej atrakcyjna, ale są też ku temu powody ważniejsze: przy rysowaniu najłatwiej natychmiast ocenić rezultat wydanego polecenia, w razie pomyłki łatwo od razu zauważyć, co się stało, a ponadto grafika jest niezbędnym elementem prawie każdego programu.

4. Czynności wykonywane wielokrotnie, procedury. Sztuka programowania to sztuka wyszukiwania w ludzkim działaniu elementów powtarzalnych. Trzeba spróbować samemu tworzyć pętlę o zadanej liczbie powtórzeń i wywołane procedury poprawiać przy pomocy edytora, budować je parametrycznie, posługiwać się funkcjami i wyrażeniami, budować struktury rekurencyjne.

5. Styl programowania: znajomość metody zstępującej od ogólnych założeń do szczegółowych rozwiązań, wyrobienie prawidłowych nawyków programowania strukturalnego: czytelnego zapisywania algorytmów, stosowania czytelnych nazw procedur i zmiennych, dobierania przejrzystych schematów współdziałania procedur, pojęcie niezmienników procedur i zależności rezultatów ich wykonania od warunków początkowych. Idzie tu o wyrobienie pewnych nawyków, uniknięcie już w zarodku wytwarzania się złych przyzwyczajeń, o zwrócenie uwagi na możliwe błędy, zanim zdążą się one utrwalić.

6. Złożone metody grafiki. Wbrew nazwie jest to sposób wprowadzenia takich podstawowych pojęć informatyki, jak struktury danych. Właśnie przy rozwiązywaniu takich problemów jak modyfikacje i przekształcenia rysunków, zmiany skali, rysowanie rysunku trójwymiarowego najłatwiej dostrzec znaczenie metod organizacji danych, różne istniejące możliwości i ich wpływ na efekt, natychmiast widoczny na ekranie.

7. Działania na tekstach. — Inny przykład struktur danych. Składanie wyrazów w zdanie i wydobywanie ich ze zdania, użycie rekurencji przy przeglądaniu i modyfikowaniu list, tworzenie zdań generowanych losowo i udzielanie prostych sensownych odpowiedzi na zdania wprowadzane z klawiatury, projektowanie prostych dialogów z komputerem wykorzystującym rozpoznawanie tekstu, operowanie poszczególnymi znakami w słowie. Przetwarzanie tekstów nie wymaga specjalistycznej wiedzy, a rezultaty są natychmiast widoczne, natomiast mamy z nimi do czynienia we wszystkich dziedzinach życia.

8. Ćwiczenia samodzielne to rozwiązywanie wybranego przez ucznia problemu oraz samodzielna analiza tego rozwiązania, wraz z zaprezentowaniem go kolegom.

Do czasu rozpoczęcia nauczania „elementów informatyki" jako normalnego już przedmiotu szkolnego pozostaje niewiele ponad rok. Do tego czasu trzeba nie tylko wyposażyć wybrane placówki w sprzęt, przeszkolić nauczycieli (Instytut Kształcenia Nauczycieli przygotowuje na rocznym kursie... 60 nauczycieli), ale także stworzyć niezbędne środowisko metodyczne nauczyciela: przykładowe programy wraz z pomocami metodycznymi, podręczniki dla ucznia i nauczyciela itp.

Publikując te założenia liczymy zarówno na dalszą krytyczną ocenę jak i na to, że autorzy różnego rodzaju kursów programowania w poszczególnych językach oraz szkoleń wezmą pod uwagę nasze idee

WŁADYSŁAW MAJEWSKI

Na razie nie widać konkurencji

Rozmowa z mgr inż. ZYGMUNTEM KORGĄ — z-cą dyrektora ds. handlowo-technicznych Zakładów Urządzeń Komputerowych Mera-Elzab w Zabrzu, produkujących mikrokomputery MERITUM I.

● Czy ma Pan swój ideał komputera osobistego?

— Oczywiście — piękny jak Macintosh, oprogramowany jak Apple i tani jak ZX-81. Tylko, że to...

● ...nie jest MERITUM.

— Tylko, że to w naszych warunkach jest niemożliwe. Ze względu na brak krajowej bazy elementowej, ani nas, ani nikogo w Polsce nie stać nawet na wytwarzanie odpowiednika SPECTRUM o porównywalnej cenie.

● Czy wobec tego nie czujecie się tu, w ELZAB-ie sfrustrowani?

— Nie. Przy tak mizernym poziomie rodzimej mikroelektroniki robimy dobry, wytrzymały mikrokomputer, obniżyliśmy dwa razy jego cenę, mamy poniżej 5 proc. awaryjności, obywamy się przy tym bez importu z Zachodu. To chyba wystarczy by mieć satysfakcję.

● Klientów wasz wyrób niezbyt jednak zachwyca. Zwłaszcza, gdy zestawiają go z Commodore albo Atari.

— I nic dziwnego. Problem jednak nie w tym co my i nabywca chcemy, bo chcemy tego samego, a w tym co i za ile można wyprodukować. Dziś budowa IBM-a PC nie wymaga superwiedzy lecz odpowiednich „kości". Nie mamy ich i tylko dlatego musimy robić MERITUM. Pretensje z tego tytułu proszę więc kierować do tych, którzy odpowiadają za rozwój technologii. Sam zresztą jestem ciekaw dlaczego mikroprocesory wytwarza się u nas w ilości zaledwie kilku tysięcy sztuk rocznie, dlaczego nie mamy własnej pamięci dynamicznej, ba — czemu każda nóżka w tranzystorze jest różnej grubości.

● Jak z tego wynika produkcja MERITUM to niemal heroizm.

— Bynajmniej. Dla ELZAB-u jest to raczej margines, naszą specjalnością są przecież monitory. MERITUM stanowi zaledwie 1 proc. produkcji zakładów.

● Po cóż więc wam ten kwiatek, skoro tyle jes' z nim kłopotów? Czyżby zysk je rekompensował?

— Finansowy z pewnością nie. Przy cenie 100 tysięcy złotych za jednostkę centralną MERITUM I, zarabiamy zaledwie 6 proc. Nasze korzyści wynikają z zaznajomienia załogi z tego rodzaju wyrobem. Ważna jest też reklama oraz zainteresowanie wzbudzane przez zakład, dzięki czemu nie mamy na przykład problemów z naborem załogi.

● A co z tego ma człowiek ciekawy mikroinformatyki?

— Jak do tej pory — kilkanaście szkolnych laboratoriów komputerowych, kilkanaście klubów o tym profilu, biuletyn dla rodziny MERITUM, możliwość zakupu polskiego personalu za polskie pieniądze.

● Bez dobrej grafiki, bez koloru, bez oprogramowania, z niewielką pamięcią i bardzo drogi.

— Cena wynika z niezbędnego importu, jest przy tym niższa niż choćby cena kolorowego telewizora. Pozostałe parametry sukcesywnie doskonalimy. Produkowany od lipca br. drugi model MERITUM I ma już pamięć dynamiczną o pojemności do 48 kB, polskie znaki alfabetyczne, przystawkę umożliwiającą współpracę z każdym czarno-białym telewizorem. Seryjnie też montujemy w nim dodatkowy port równoległy dedykowany pamięci zewnętrznej na dysku elastycznym. Zgodnie z zapowiedziami we wrześniu ruszyła produkcja wersji II, sprzedawanej w zestawach z NRD-owską stacją 5-calowych dyskietek.

● A grafika?

— W przyszłym roku będziemy sprzedawali na życzenie zamawiającego modele z dodatkową płytką, która pozwoli uzyskiwać dużą rozdzielczość punktową. Poza tym oferujemy oprogra-

MERITUM

Podstawowe dane techniczne:

Pamięć operacyjna	17 kB
Pamięć stała	14 kB
Pamięć obrazu	1 kB

Układy sterowania monitorem ekranowym bądź standardowym odbiornikiem telewizyjnym:
- organizacja obrazu: 16 × 64 znaki lub 16 × 32 znaki wybierane z klawiatury
- reprogramowany generator znaków
- semigrafika
- sygnały wyjściowe: zespolony sygnał wizyjny, sygnał wyjściowy w. cz. zmodulowany

Interfejsy: szeregowy wg standardu RS232C, 3 interfejsy rownolegle we/wy (z możliwością indywidualnego definiowania linii sterujących).
Klawiatura kontaktronowa typu QWERTY obsługiwana programowo.

OPROGRAMOWANIE

Oprogramowanie podstawowe mikrokomputera MERITUM-I stanowi 12-kbajtowy rezydujący język BASIC-MERITUM uzupełniony o moduły zarządzające i obsługujące klawiature, porty, we/wy wyświetlane oraz interfejsy zewnętrzne.
Komputer może pracować w czterech trybach pracy:
- Tryb bezpośredni (kalkulatorowy)
- Tryb edycji
- Tryb wykonywania programu
- Tryb systemowy.

(r.p.)

mowanie. Głównie narzędziowe (np. Asembler dla Z-80, Pascal, oraz użytkowe jak choćby baza danych). Chcemy, choć nie jest to nasze zadanie, kupować i rozprowadzać również inne programy. Na razie jednak nie ma dla takich działań podstaw prawnych.

● Czy to wszystko, czego może się od was spodziewać użytkownik MERITUM?

— Na razie niczego więcej nie planujemy.

● Jak długo zamierzacie więc produkować te urządzenia?

— Dopóki będzie na nie zbyt. Rocznie możemy dać na rynek do 4 tysięcy sztuk.

● Nie boi się Pan utraty klientów?

— To jest pytanie o konkurencję, której na razie nie widać. A szkoda.

Rozmawiał: KRZYSZTOF KRUPA

CO ROBIĄ INNI

ROZMACH I DETERMINACJA

Komputery osobiste w ZSRR

„Musimy uruchomić program na podobieństwo tego, jaki podjęliśmy po Rewolucji Październikowej w celu likwidacji analfabetyzmu" — tak prof. ANATOLIJ ALEKSANDROW, prezes Akademii Nauk ZSRR określa wagę i rozmach programu powszechnej komputeryzacji podejmowanego właśnie w Związku Radzieckim.

W pierwszym rzucie zamierza się wprowadzić ponad milion komputerów osobistych do systemu oświatowego oraz przeszkolić — jak pisała „Prawda" — „wielką armię nauczycieli i zarządców gospodarki". Nauka posługiwania się komputerami rozpocznie się w szkołach średnich już od jesieni br.

Decyzje podejmowane w tym zakresie — organizacyjne, produkcyjne, personalne — świadczą o olbrzymiej determinacji w zwalczaniu „analfabetyzmu mikrokomputerowego" w ZSRR. Jest to w pełni zrozumiałe, biorąc pod uwagę zapoczątkowany zwrot w gospodarce radzieckiej, w którym postęp naukowo-techniczny spełniał będzie rolę kluczową.

— Wkrótce zaczniemy liczyć komputery w milionach sztuk — stwierdził w lipcu br. na łamach „Literaturnej Gaziety" prof. Jewgienij Wielichow, 54-latek, wiceprezes Akademii Nauk ZSRR, najbardziej chyba kompetentny człowiek w zakresie wprowadzania techniki mikrokomputerowej w Związku Radzieckim. — Mamy ku temu wszystkie możliwości techniczne. Trzeba tylko dokładnie przemyśleć aspekty organizacyjne tej sprawy. Usunąć bariery, które krępują i powstrzymują ruch naprzód. (...) Przegrupowanie na tych wszystkich kierunkach już u nas trwa. Być może nie zawsze z optymalnymi rezultatami. Ale uczymy się na swoich błędach, szybko poprawiamy je, z powodzeniem pokonujemy główną przeszkodę na drodze najnowszej techniki — barierę psychologiczną. Wszyscy powinni zdać sobie sprawę z tego, że era komputerów już nastąpiła i postawiła swoje wymagania przed każdym z nas.

Radziecki program komputeryzacji oświaty i gospodarki wywołuje zrozumiałe zainteresowanie za granicą. Jak twierdzi „Wall Street Journal" 26 zachodnich firm elektronicznych otworzyło w ostatnich miesiącach swoje przedstawicielstwa w ZSRR, a dwa razy tyle zamierza to uczynić licząc na ewentualne kontrakty. Przypomina się w tym kontekście, że podczas ubiegłorocznej, grudniowej wizyty Michaiła Gorbaczowa w Wielkiej Brytanii, znajdujący się w składzie delegacji akademik Jewgienij Wielichow spotkał się m.in. z przedstawicielami ICL i innych komputerowych firm brytyjskich.

Wszyscy są jednak zgodni, że — niezależnie od zakupów za granicą — Związek Radziecki oprze swój program rozwoju na mikrokomputerach własnej produkcji. Jak stwierdził niedawno akademik Gurij Marczuk, matematyk, przewodniczący Państwowego Komitetu ds. Nauki i Techniki: **„Twierdzenia jakoby postęp ZSRR w dziedzinie elektroniki był uzależniony od dostępu do zachodnich technologii brzmią jak naiwne oszukiwanie samego siebie lub w najlepszym wypadku jak pomyłka".**

Pierwszym radzieckim komputerem osobistym jest 8-bitowy „Agat", o wydajności 300 tysięcy operacji na sekundę, z pojemnością pamięci operacyjnej do 256 kilobajtów, pamięci stałej do 32 kilobajtów, z wyprowadzeniem informacji na taśmę papierową i monitor. Właśnie „Agat" stanie się na początku podstawowym mikrokomputerem wprowadzanym do szkół.

Kolejna generacja to 16-bitowa „Elektronika BK-0010", mająca szansę stać się w przyszłości standardowym wyposażeniem radzieckich szkół (obecnie kosztuje 840 rubli, a po pełnym uruchomieniu produkcji 550...600 rubli).

Trwają prace nad pokonaniem następnego progu. Jak stwierdzono podczas narady w Akademii Nauk ZSRR: „Przejście do produkcji 32-bitowych komputerów będzie wymagało utworzenia nowej bazy podzespołów. Trzeba rozwiązać zadanie standaryzacji tej bazy". Niezbędnym jest również — stwierdzono — uruchomienie własnej produkcji różnych typów pamięci, monitorów i różnych urządzeń współpracujących. Finalizowane obecnie plany współpracy ZSRR z innymi krajami socjalistycznymi w ramach RWPG sięgające horyzontu roku 2000, mają właśnie głównie na względzie rozwój i standaryzację bazy podzespołowej elektroniki.

Wyścig mikroprocesorowy trwa...

WALDEMAR SIWIŃSKI

HP-110 walizkowy komputer Hewlett-Packarda



MICRO-COMPUTER '85

Korespondencja z Frankfurtu

„Micro — computer'85" — to nazwa nowej imprezy handlowo-technicznej, targów, jakie odbywac się będą już corocznie we Frankfurcie nad Menem. Takich wyspecjalizowanych targów dotąd w RFN nie było, choć oczywiście mikrokomputery eksponowano na wszystkich większych wystawach sprzętu elektronicznego.

RFN wydaje się być obecnie workiem bez dna dla producentów, zwłaszcza tanich mikrokomputerów. Renomowany instytut koniunktur Diebolda ocenia, że na komputery domowe (w cenie poniżej 1500 DM) od kilku lat utrzymuje się boom. Na początku roku 1983 komputerków takich było w RFN 197 tys. sztuk, rok później 707 tys., a na początku bieżącego roku 1,2 mln sztuk. Do końca bieżącego roku ich liczba ma osiągnąć 1,8 mln. W roku 1990 rynek mikrokomputerowy ocenia się na 3,75 mln sztuk.

We frankfurckich targach wzięło udział ponad 170 firm z 10 krajów, łącznie z najpoważniejszymi producentami amerykańskimi i japońskimi. Był to więc miarodajny przegląd tego co w technice mikrokomputerowej liczy się lub będzie się liczyć na rynku zachodnioeuropejskim.

Handlowcy i producenci dzielą dziś mikrokomputery na dwie podstawowe grupy: PC (personal computer) i HC (home computer — komputery domowe).

W grupie PC firma Apple prezentowała swoje modele IIc, IIe oraz Macintosh i Lisa. IBM reklamowała swoją rewelację IBM AT. Ten komputer dzięki procesorowi Intel 80286 i nowej architekturze wewnętrznej jest 2—3 razy szybszy od poprzednich. Będzie więc poważną konkurencją dla Appla II.

Techniczną nowością roku 1984 i 1985 są walizkowe komputery osobiste. Oferuje je już kilkadziesiąt firm. Wyposażone są w ciekłokrystaliczne lub luminescencyjne płaskie ekrany, zasilane są z akumulatorów lub baterii i ważą od jednego do kilku kilogramów.

Mimo niewielkich rozmiarów mają możliwości dorównujące stacjonarnym komputerom osobistym. Demonstrowany po raz pierwszy jesienią 1984 r. walizkowy model Hewlett — Packarda oznaczony numerem HP—110 pracuje na procesorze 16-bitowym 80C86, zawiera pamięć RAM 272 kB i ROM 384 kB. Ma odkładany nad klawiaturą ciekłokrystaliczny ekran zawierający 128 × 480 punktów, co pozwala zapisać 16 wierszy po 80 znaków. Wymiary 33 × 25,5 × 7,5 cm, waga — 4,5 kg. Kosztował na początku bieżącego roku prawie 3 tys. dolarów. Nie jest to więc zabawka, chociaż fachowcy mają np. zastrzeżenia, że oprogramowanie do obróbki tekstów jest zbyt uproszczone. HP—110 nie potrafi bowiem dzielić i przenosić wyrazów, przez to na prawym marginesie tekstu napisanego na tym komputerze powstają nieestetyczne luki, co pedantów podobno razi.

Osobiste komputery walizkowe to nowy, poważny biznes na rynku mikrokomputerowym, więc łapie się za nie kto żyw. Nawet Philips, znany dotąd bardziej z maszynek do golenia niż z komputerów wyprodukował prototypowy walizkowy PC. Dość duża to wprawdzie walizka — waży 15 kilogramów, więc do noszenia na co dzień nie bardzo się nadaje (chyba że nosiłby specjalny bagażowy), ale ma wszystko, co szanujący się komputer osobisty mieć powinien. P2000C dysponuje pamięcią dyskową o pojemności 640 kB, ma wbudowane programy do obróbki tekstów oraz sporządzania wykresów i tabel. Wspominam ten zupełnie chyba nieznany produkt, aby pokazać jak wielkie zainteresowanie w przemyśle elektronicznym nadal budzą mikrokomputery. W RFN, jak policzyłem, na początku 1985 r. komputery osobiste produkowało 55 firm, w tym takie, których nie podejrzewałoby się o takie zainteresowania. Np. znany koncern chemiczny BASF opracował trzy modele PC.

Świadczy to oczywiście o tym, że szanse w tej branży wciąż jeszcze mają wchodzący na rynek nowi producenci, a także, że nie ma jeszcze takiej konstrukcji, ,o której byłoby wiadomo, że podbije rynek światowy.

Dość duża ewolucja techniczna następuje także w grupie komputerów domowych. Na frankfurckich targach przebojami w tej grupie były modele znajdujące się właściwie na granicy cenowej i technicznej komputerów osobistych. Handlowcy i producenci uważają za komputer domowy urządzenie, którego cena nie przekracza 500 dolarów (w RFN — 1500 marek). I w tych granicach mieści się zarówno nowy komputer Sinclaira QL jak i Commodore plus/4. A ten ostatni jest właściwie już zupełnie przyzwoitym komputerem osobistym, którego można używać profesjonalnie, ma bowiem wbudowane programy optymalizacyjne, kalkulacyjne i graficzne. Model QL natomiast jest pierwszym w tej klasie i w tym przedziale cen mikrokomputerem pracującym na 32-bitowym procesorze.

Ciekawostką techniczną, ale — jak mi się wydaje bez większej przyszłości handlowej — jest japońska rewelacja firmy Seiko, reklamowana jako „pierwszy na świecie komputer w ręcznym zegarku". W istocie jest to dość prościutki komputer wielkości kalkulatora naukowego, w którym zegarek spełnia funkcję ekranu.

Seiko produkuje dwa systemy tego typu. Jeden składa się z zegarka UC-2000 (zawierającego 4-bitowy procesor, pamięć RAM 2 kB, ROM 6 kB, ekran ciekłokrystaliczny mieszczący 4 wiersze po 10 znaków w każdym) i urządzenia sterującego UC-2200 (zawierającego procesor Z-80, pamięć RAM 4kB, ROM 26 kB, klawiaturę z 70 przyciskami i termodrukarkę). Drugi, uproszczony system składa się z zegarka i klawiatury. Pozwala on w samej pamięci zegarka umieścić 2000 znaków, programowanych w języku Basic. Mogą to być numery telefonów, adresy albo rozkład jazdy. Finezyjną (i opatentowaną) nowinką techniczną jest w tej konstrukcji bezprzewodowe, indukcyjne połączenie zegarka z innymi elementami systemu — po prostu wkłada się go w odpowiednie zagłębienie i to wystarcza do niezawodnego przekazywania impulsów.

Nie sądzę, aby na tym rozwiązaniu Seiko zrobić mogła wielki interes, ale zwracam na tę ciekawostkę uwagę Czytelników, bowiem obrazuje ona możliwości dalszej niebywałej miniaturyzacji komputerów. A być może także nowy trop wiodący w przyszłość, ,mianowicie zamiast dużego komputera osobistego będziemy nosić ze sobą tylko miniaturową końcówkę czyli terminal połączony z centralą radionadajnikiem.

JAN RURAŃSKI



ATARI ST — pościg za Macintoshem

„Komputer domowy? Nigdy o czymś takim nie słyszałem! Ja produkuję komputery personalne" — oświadcza z rozbrajającym usmiechem na twarzy buldoga **Jack Tramiel**, jedna z legendarnych postaci rynku mikrokomputerowego, założyciel Atari, potem twórca sukcesów Commodore, który ostatnio powrócił do swej macierzystej firmy. Jeśli zdoła przemienić swe słowa w czyn, rok 1985 będzie ostatnim, w którym ktokolwiek będzie chciał w ogóle słuchać o komputerach domowych.

Wprowadzając bowiem na rynek nową serię Atari ST podjął on probę zawarcia mocy obliczeniowej mikrokomputerów personalnych i wygody we współpracy z użytkownikiem typowej dla maszyn takich jak Macintosh, w urządzeniu o cenie poniżej 500 dol.

Równocześnie Atari uczyniło swoją tradycyjną rodzinę komputerków 8-bitowych o wiele poważniejszą dzięki wzbogaceniu firmowego oprogramowania oraz wyposażenia. Obecną cenę (250 funtów za Atari 800 XL z 64 kB RAM wraz ze stacją dyskietek i programem zarządzającym) z pewnością można uznać za atrakcyjną. Na tym tle nieco mniej dziwi optymizm Tramiela, który w okresie, w ktorym Commodore po raz pierwszy wykazuje spadek sprzedaży, Sinclar musiał sprzedać swój komputerowy interes, Acorn ubiega się o ratunek u Olivetti a mniejsze firmy, takie jak Oric padają — dumnie obwieszcza, że zamierza w tym roku wyprodukować i sprzedać rekordową ilość pięciu milionów egzemplarzy komputerów Atari twierdząc, że na rynku zawsze znajdzie się miejsce na właściwy produkt po właściwej cenie. Spokojnie też wzrusza ramionami, gdy rynkowi rywale głoszą, że nie będzie on w stanie wyprodukować urządzenia zapowiadanej jakości bez opóźnień i bez konieczności podnoszenia ceny. Zapytany o sceptyczny komentarz sir Clive Sinclaira na temat nowego Atari Tramiel zapytał ironicznie: Kto to taki ten Sinclair?

Nowa seria Atari wyposażona jest — wzorem Macintosha — w tzw. mysz, przesuwane po stole pudełeczko, którego ruchy natychmiast odwzorowywane są przez wskaźnik na ekranie. Dzięki rozbudowanemu oprogramowaniu (Digital Research Graphics Environment Manager) mysz jest praktycznie podstawowym środkiem kontaktu użytkownika z maszyną, znacznie szybszym i wygodniejszym od klawiatury. Choć mysz kojarzy się wszystkim z Macintoshem, Tramiel odrzuca zarzuty naśladownictwa twierdząc, że idea została sformułowana po raz pierwszy 10 lat temu w pracowniach Xeroxa w Palo Alto.

Oprogramowanie współpracujące z myszą w Atari nie wydaje się uboższe niż w Macintoshu (mimo kilkakrotnej różnicy ceny obu urządzeń). W dodatku Macintosh dysponuje jedynie ekranem czarno-białym, podczas gdy Atari wkracza na rynek pod hasłem „życie nie jest czarno-białe" i oferuje obok czarno-białego trybu pracy o najwyższej rozdzielczości (640 na 400 punktów) tryb pośredni (640 na 200 punktów) z czterema kolorami i tryb o niskiej rozdzielczości (320 na 200 punktów), w którym na ekranie można równocześnie używać 16 kolorów.

Atari ST nie ustępuje również Macintoshowi pamięcią i mikroprocesorem. Jego sercem jest, podobnie jak w Macintoshu, pełna wersja Motoroli 68000 z 16-bitową szyną danych — co daje niewątpliwą przewagę np. nad Sinclairem QL wyposażonym w ten sam mikroprocesor z ośmiobitową szyną. Nawet najtańszy model nowej serii Atari — 130 ST wyposażony będzie w 128 kB RAM oraz 192 kB (!) ROM, mieszczące całe oprogramowanie użytkowe myszy oraz Basic lub Logo — do wyboru czy, za niewielką dopłatą oba języki. Dodatkowo do maszyny podłączyć można kartridż (podłączaną osobną kostkę z pamięcią ROM) o pojemności 128 kB. Firma obiecuje duży wybór oprogramowania na kartridżach. Nowy Atari podsiada wbudowany interfejs do stacji dyskietek oraz port dla 15 MB Winchestera, który ma kosztować ok. 500 dol.

Tramiel podkreśla, że pojawienie się twardego dysku w cenie dostępnej dla kieszeni przeciętnego odbiorcy oznacza domową rewolucję i przemianę komputera w rodzinny bank danych. Na takim dysku można pomieścić — dla przykładu — całe ustawodawstwo brytyjskie z okresu ostatnich 200 lat.

Standardowym wyposażeniem nowych rodzin mikrokomputerów jest 3-kanałowy generator dźwięku i szumów z możliwością sterowania wysokością, brzmieniem i głośnością tonów oraz komplet wyjść: Centronics — dla drukarki, RS 232, joystick, TV, RGB, composite video.

Cena modelu 130 ST wynosi ok. 450 dol. (wraz z myszą, podczas gdy 520 ST z 512 kB RAM kosztować ma ok. 650 dol., przy czym w pierwszym okresie sprzedaży jest on wyłącznie w zestawach obejmujących napęd dyskowy, drukarkę i monitor — całość za 1000 funtów (ok. 1300 dol.). Ceny te wyglądają na niewiarygodnie niskie, ale Tramiel z naciskiem oświadcza: „Nie wiem co rozumiecie przez słowo tani, gdyż będzie to produkt wysoce dochodowy — każdy wprowadzony dotąd przeze mnie komputer przynosił zyski". W ustach promotora Commodore 64 słowa te brzmią wiarygodnie.

Zarówno Sinclair jak i Apple przekonały się na własnej skórze, że firmy softwarowe potrzebują około roku na podjęcie produkcji oprogramowania dostosowanego do nowo wprowadzonego na rynek 16-bitowego modelu. Tramiel deklaruje że przekazał oprogramowanie systemowe serii ST 25 największym firmom softwarowym na pół roku przed wejściem na rynek, tak więc pierwsze programy będzie można kupić równocześnie z pierwszymi urządzeniami

wg YOUR COMPUTER

DEMON GRY

Rodzice w czasach naszej młodości fascynowali się kolejką elektryczną. Bawili się nią wspólnie z nami, był to powrót do młodości fascynacja zabawkami nieosiągalnymi. Dzisiaj kolejka elektryczna co prawda nadal cieszy, ale powoli wypierana jest przez komputery które oprócz swoich możliwości użytkowych są nową zabawką XXI wieku.

W ciągu ostatnich dwóch lat dokonał się ogromny postęp w rozwoju gier komputerowych. Powstały setki firm specjalizujących się w przygotowywaniu i projektowaniu coraz atrakcyjniejszych gier. Rozwój przebiegał w dwóch zasadniczych kierunkach — polepszenia grafiki i fabuły. Został przyjęty następujący podział typów gier.

1. gry zręcznościowo-sprawnościowe (Arcade),
2. przygodowo-tekstowe (Adventure),
3. graficzno-przygodowe (Graphic Adventure),
4. strategiczne (Strategic),
5. symulacyjne (Simulation)

Można dokonywać bardziej szczegółowego podziału w zależności od tematyki, np. symulacyjne — handlowe, kiedy to bawimy się w zarządzającego jakąś firmą (musimy podejmować decyzje co kupić, lub sprzedać a komputer ocenia nasze przedsięwzięcia, w efekcie czego bankrutujemy albo się rozwijamy). Przykładem takiej gry jest program „linie lotnicze". Do innego typu należą wszelkiego rodzaju symulacje kierowania czy też pilotowania samolotów, helikopterów, statków kosmicznych). Uczymy się startować, lądować, latać według mapy, w niektórych grach wprowadzone są nawet elementy walki, jak chociażby w programie „Fighter pilot"

Programy, które oferują nam firmy są coraz atrakcyjniejsze. Występują oczywiście ograniczenia, takie jak pojemność pamięci, rozdzielczość obrazu, dźwięk. Jest to związane z rodzajem sprzętu, jaki mamy do dyspozycji. Jednak i tu możliwości zwiększają się, gdyż rynek sprzętowy rozwija się dynamicznie.

Rok temu pojawiły się pierwsze gry oparte o systemach laserowych. Mając do dyspozycji joystick i komputer sprzężony z dyskiem laserowym, na którym zapisany jest program — film rysunkowy, możemy kierować akcją tej projekcji poprzez oddziaływanie na postac bohatera. Fabuła może być następująca: zły smok porwał księżniczkę, jesteś rycerzem, który chce ją uwolnić i pokonać smoka. Obraz jaki widzimy na ekranie jest jak w filmach Walta Disneya.

Zaczynamy grę: nasz rycerz stoi przed wejściem do lochów. Dając sygnał do przodu powodujemy, iż postać na ekranie wchodzi w czeluść i w tym momencie zaczynają się niebezpieczeństwa: zapada się podłoga w zależności od tego jak poruszymy joystickiem, tzn. czy zrobimy ruch właściwy czy nie, komputer wybierze odpowiednią sekwencję filmu. W tym konkretnym programie, jeśli nie uda nam się w odpowiednim momencie wyskoczyć na pomost, komputer zrealizuje program katastrofy, zobaczymy jak nasz rycerz rozbija się na dnie przepaści.

W tym przypadku mamy do czynienia z symulacją rysunkową, ale możliwe są i symulacje filmów realistycznych. Jak na razie jesteśmy przywiązani do ekranu telewizyjnego, jednak już dzisiaj wiele ośrodków pracuje nad tak zwaną projekcją przestrzenną. Można sobie wyobrazić, że już w bardzo niedalekiej przyszłości projekt ten zostanie zrealizowany, co pozwoli na jeszcze większe urealnienie świata bajkowego, w jakim będziemy przeżywać przygody naszych bohaterów.



Istnieją jednak pewne niebezpieczeństwa zbytniej realności takiej zabawy. Należy zwrócić uwagę na poważne konsekwencje wynikające z nieprawdopodobnej sugestywności owych bajek przyszłości, a co wiąże się z tym faktem, możliwości całkowitego oderwania się grających od otoczenia. Nie jest to problem wydumany, świadczą o tym doświadczenia socjologiczne oddziaływania współczesnych, „prymitywnych" jeszcze gier komputerowych.

JACEK BODEK

Matchbox dla szofera

DOKOŃCZENIE ZE STR. 4

Polsce pierwszy (i ostatni!) kontakt z informatyką następuje poprzez „zabawki komputerowe". Zaczyna się utożsamianie informatyki z tymi zabawkami, musimy więc liczyć się z różnego rodzaju, zupełnie naturalnymi w tej sytuacji, odruchami społecznymi. Po pierwsze: czy nas na to stać? Bo jeśli to jest tylko zabawa, to czy naprawdę trzeba na to wydawać dolary, skoro nie ma strzykawek jednorazowych? Następna sprawa — pojawia się pytanie czy to nie jest przypadkiem zjawisko dekadenckie (oni już „tam" są tacy bogaci, że nie mają nic lepszego do roboty, to się tym bawią).

Zastosowania zabawowe w naszej obecnej sytuacji nie pełnią też roli wychowawczej czy przygotowawczej, jak w społeczeństwach uprzednio zinformatyzowanych. W krajach wysoko rozwiniętych, rozrywka informatyczna jest między innymi środkiem na przezwyciężenie poważnych resentymentów społecznych, związanych z podziałem na „tych", którzy wiedzą i „tych", co nie wiedzą. Danie takich zabawek w cenie kilku kartonów papierosów znakomicie odcudowniło informatykę, natomiast w Polsce... Pomyślmy, co będzie się działo z młodym człowiekiem, który opanował trudną sztukę klepania w klawisze. Jest to umiejętność do niczego niepotrzebna, bez dalszego ciągu w pracy zawodowej, a przez to — ogromny powód do frustracji.

Z drugiej strony, obawiam się, że decydenci, których gryzie sumienie, że nie jest klawo z tą naszą informatyką, postanowią bardzo tanim kosztem „odfajkować" sprawę. To znaczy przez zabawkowe komputery „rozwiązać" problem informatyki.

● **Są więc niebezpieczne aspekty tej mody.**

— Jest jeszcze jedno kuriozalne zjawisko: duże zainteresowanie naszych naukowców tymi zabawkami. Fakt ten pokazuje, jak bardzo nieprzystępne były dotychczasowe środki informatyczne w uczelniach i instytutach. Nikt mi nie wmówi, że nasze ODRY potrafią mniej policzyć niż SPECTRUM. Na każdej ODRZE, od 1204 w górę (a 1204 była zbudowana 20 lat temu!) można policzyć więcej niż na SPECTRUM. Nie mówię już o R-32, których możliwości obliczeniowe są nieporównanie większe niż IBM PC, nawet wariantu XT. Natomiast dostępność naszych uczelnianych i instytutowych komputerów była tak mała a przyjęte metody pracy z nimi — tak odstraszające, że uczeni zajęli się zabawkami!

Współczesne, duże komputery są niemniej „przyjazne" dla użytkownika niż osobiste. Ma on na biurku tak samo wyglądający ekran, taką samą klawiaturę, lecz za tym stoi potężna moc obliczeniowa i ogromny bank informacji, którego nie można wsadzić do komputera o architekturze ośmio czy szesnastobitowego „personala". (To nie tylko kwestia pojemności pamięci winchesterowych nawet dysków, lecz ograniczoność dostępu do pamięci przez bardzo mizerne możliwości maleńkiego procesora). Nasi profesorowie natomiast, skutecznie zniechęceni przez centra obliczeniowe, usiłują posługiwać się sprzętem takim, jakim byłby „matchbox" dla taksówkarza plastikowa szabelka dla żołnierza czy teatralna lornetka dla astronoma.

● **Jak w takim razie zachować się wobec faktu wciąż przecież narastającego zainteresowania „małą" informatyką? Jakie widzi Pan Profesor zadania dla naszego BAJTKA?**

— Dżentelmeni nie obrażają się na fakty. Nie wykluczone, że można tę popkulturę ukierunkować w stronę pozytywistyczną, zaprzęgnąć te zabawki do edukacji, do jakiegoś użytku nie wyzutego z walorów intelektualnych.

● **Uczyć programowania?**

— Nie jestem przeświadczony, że umiejętność programowania jest powszechnie potrzebna. Tym bardziej że jest to sztuka bardzo trudna — jeśli mamy na myśli rzeczywiście programowanie, a nie składanie elementarnych „programików".

Warto natomiast uczyć umiejętności poprawnego formułowania problemów, efektywności ich rozwiązywania a także pewnych prawd, które obowiązują przy ścisłym rozumowaniu. Warto np. wykazywać, że istnieją zadania nierozwiązywalne (nie takie, których nie umiemy rozwiązać, lecz takie które nie mają rozwiązania), że źle postawione zadanie dopuszcza byle jakie odpowiedzi. Należy uczyć, że dla ścisłego rozumowania jest niezbędne skrupulatne przestrzeganie wstępnie przyjętych konwencji, że ich zmiana w trakcie toczącego się procesu przetwarzania informacji nieuchronnie prowadzi do nonsensu.

Jest to wiedza uniwersalna, której naszemu społeczeństwu brakuje. Nazwałbym to sztuką rozumowania algorytmicznego. Dość łatwo można ją praktykować nawet z prostymi i tanimi komputerkami, choć niezbyt przydatne okażą się przy tym te pozornie bardzo łatwe do opanowania języki programowania, używanie których zbyt często prowadzi do nieokreślonych, a więc arbitralnie przez komputer znajdowanych rozwiązań.

● **Co jeszcze można osiągnąć wykorzystując komputerową modę?**

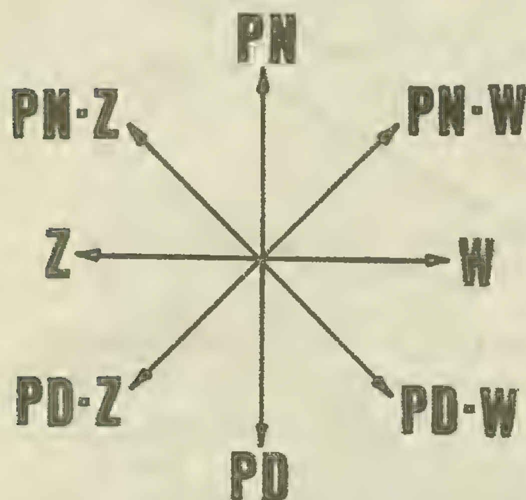
— Chyba niewiele więcej. Z drugiej jednak strony, gdyby udało się rozpropagować umiejętność ścisłego rozumowania już to samo byłoby niezwykle wartościowym osiągnięciem.

W rozmowie uczestniczyli:

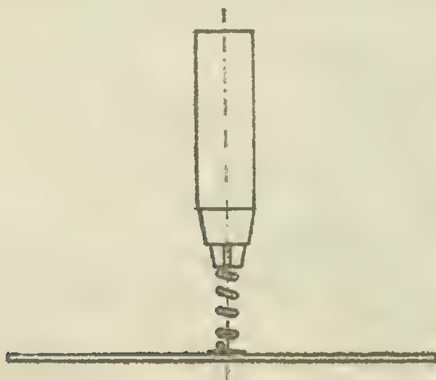
ANDRZEJ BRZEZICKI
WŁADYSŁAW MAJEWSKI
ROMAN POZNAŃSKI
WALDEMAR SIWIŃSKI

Drążek sterowy

Akcją gry komputerowej można sterować za pomocą klawiatury. Jest to jednak sposób posiadający dwie zasadnicze wady: zużywanie się klawiatury i mała szybkość reakcji. Niedoskonałości tych można uniknąć, stosując sterowanie za pomocą tzw. drążka sterowego (ang. joystick).



Rys. 1. Kierunki ruchu możliwe do uzyskania za pomocą drążka sterowego



Rys. 2. Konstrukcja przegubu

Drążek sterowy przekazuje do komputera ruchy ręki gracza i pozwala całkowicie skoncentrować się na obrazie telewizyjnym. Posługiwanie się nim jest bardzo proste, wychylenie drążka w odpowiednim kierunku powoduje identyczny ruch kierowanego obiektu na ekranie telewizora. Ruch ten może odbywać się w ośmiu podstawowych kierunkach (rys. 1). Oprócz tego każdy drążek sterowy posiada osobny przycisk pozwalający na włączanie dodatkowych czynności.

Ponieważ drążek sterowy zazwyczaj nie wchodzi w podstawowe wyposażenie mikrokomputera, podajemy sposób zbudowania go we własnym zakresie. Rozwiązanie zastosowane przeze mnie jest jednym z możliwych wariantów, a jego zaletą jest prostota konstrukcji. Do budowy drążka sterowego

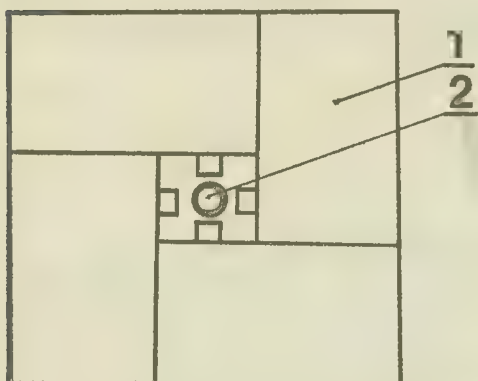
potrzebne Ci będą:

- obudowa o wymiarach pozwalających na wygodne uchwycenie w dłoni (np. obudowa radiotelefonu)
- pięć miniaturowych, astabilnych przełączników (np. PZG 3 lub DZG 1)
- rękojeść
- płytka montażowa
- gęsto nawinięta, mocna sprężyna o możliwie małej średnicy zwoju
- przewód sześciożyłowy
- odpowiednie gniazdko służące do połączenia drążka sterowego z komputerem (np. Eltra 881).

Po zgromadzeniu ww. elementów możemy przystąpić do montażu.

Konstrukcja przegubu

Istotą rozwiązania jest zastosowanie jako przegubu i równocześnie mechanizmu cofającego drążek do położenia neutralnego jednej, pionowo osadzonej sprężyny (podobnie jak zamocowane są figurki graczy w popularnym „footballu" dla dzieci). Sprężynę możemy zamocować na przyklejonej (klejem o dużej wytrzymałości mechanicznej np. „Epidianem") do płytki montażowej pinezce lub śrubie przykręconej do płytki. Otrzymujemy w ten sposób sprężysty przegub, do którego możemy zamocować rękojeść. W mojej konstrukcji wykorzystałem ołówek automatyczny. Średnica sprężyny była na tyle mała, że mogłem ją wsunąć w otwór służący normalnie do mocowania grafitu, otrzymując mocne osadzenie rączki (rys. 2). Można wykonać rękojeść z odpowiednio wyprofilowanego drewna i w dolnej części wywiercić otwór,



Rys. 3. Rozmieszczenie przełączników. 1 — przełącznik, 2 — drążek

do którego należy wkręcić wolny koniec sprężyny.

Kolejnym etapem pracy będzie

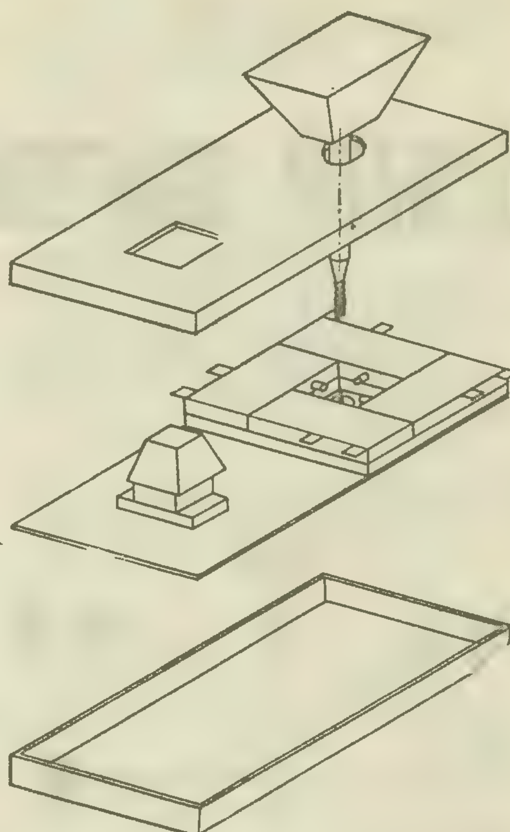
mocowanie przełączników

Ich układ przedstawia rys. 3. Muszą być ustawione w ten sposób, by niewielkim ruchem drążka można było uruchomić bądź jeden z przełączników, bądź też równocześnie dwa sąsiadujące ze sobą. Aby można było wygodnie i precyzyjnie manipulować drążkiem, przełączniki należy umieścić nieco ponad płytką montażową na kawałku sklejki. Sposób zamocowania przełączników wraz z dodatkowym przyciskiem pokazano na rys. 4. Na tym samym rysunku przedstawiono również montaż płytki z przełącznikiem w obudowie radiotelefonu „Trop". W górnej części obudowy należy wywiercić odpowiednie otwory na drążek sterowy i przycisk. Pozostaje nam już tylko

wykonanie połączeń elektrycznych

Część mikrokomputerów posiada specjalne wtyki umożliwiające przyłączenie jednego lub dwóch drążków sterowych (np. Atari, Commodore), do pozostałych niezbędny jest odpowiedni interfejs. Standardowy wtyk połączeniowy zawiera 9 bolców, z których 6 wykorzystujemy do sterowania (rys. 5). Przesunięcie obiektu w wybranym kierunku osiągamy zwierając odpowiednie wyjścia z masą (oczywiście dodatkowo musimy dysponować odpowiednim programem) np.: zwierając wyjścia 1 i 8 otrzymujemy przesunięcie pionowe w górę (północ), a zwierając wyjścia 3 i 8 otrzymujemy przesunięcie poziomo w lewo (zachód). Kierunki pośrednie (o ile zostały uwzględnione w programie) otrzymujemy zwierając równocześnie dwa wyjścia z masą, np. — zwarcie wyjść 1 i 3 z wyjściem nr 8 daje nam przesunięcie na ukos w lewo i w górę (północny zachód).

Schemat połączeń elektrycznych układu przedstawia rys. 6. Przełączniki I, II, III i IV służą



Rys. 4. Drążek sterowy, sposób montażu

do poruszania obiektu po ekranie (w kierunkach zaznaczonych na rysunku) a przełącznik nr V — do włączania innych czynności.

Po wykonaniu połączeń możemy przystąpić do przetestowania drążka sterowego. Poniżej podaję przykład programu na mikrokomputer Commodore VC-20 umożliwiający sprawdzenie działania naszego urządzenia. Po podłączeniu drążka sterowego na ekranie będą pojawiać się cyfry odpowiadające poszczególnym położeniom drążka:

położenie neutralne — 8
północny zachód — 7
zachód — 6
południowy zachód — 5
południe — 4
południowy wschód — 3
wschód — 2
północny wschód — 1
północ — 0

W momencie naciśnięcia przycisku "ogniowego" na ekranie pojawi się napis "STRZAL":

```
Ø DIM SK (2,2): D1 = 37139 : D2 = 37154
1 POKE D1, Ø
2 PA = 37137 : PB = 37152
3 FOR I = Ø TO 2 : FOR J = Ø TO 2 : READ
SK (J,I) : NEXTJ,I
4 DATA 7,0,1,6,8,2,5,4,3
```

```
1Ø GOSUB 95ØØ
2Ø CC = SK (x + 1, y + 1)
3Ø PRINT CC
4Ø IF FR = 1 THEN PRINT "STRZAL"
5Ø GOTO 1Ø
6Ø :
95ØØ POKE D2,127 : S3 = — (PEEK(PB)AND 128) =Ø) : POKE D2,255
951Ø P= PEEK(PA) :S1= — ((PAND 8) =Ø) :S2)= ((PAND 16)=Ø) :SØ= ((PAND 4)=Ø)
952Ø FR= — ((PAND 32)=Ø) : x = S2+S3 : y=S1+SØ: RETURN
```

Dobrej zabawy!

TOMASZ LISON

Od redakcji

Popularny w Polsce ZX SPECTRUM nie ma — niestety — możliwości bezpośredniego podłączenia drążka sterowego, niezbędne jest więc zastosowanie interface'u. Pozwala on na współpracę komputera z dwoma drążkami.

Programy dla joysticków bywają zwykle układane z zastosowaniem kodu maszynowego, jednakże istnieje możliwość pisania ich w BASIC-u. Drążki spełniają identyczne funkcje co najwyższy rząd klawiszy. Zależność jest następująca:

klawisz 1 = drążek 2 w lewo
klawisz 2 = drążek 2 w prawo
klawisz 3 = drążek 2 w dół
klawisz 4 = drążek 2 w górę
klawisz 5 = drążek 2 przycisk dodatkowy
klawisz 6 = drążek 1 w lewo
klawisz 7 = drążek 1 w prawo
klawisz 8 = drążek 1 w dół
klawisz 9 = drążek 1 w górę
klawisz 10 = drążek 1 przycisk dodatkowy

Pozycja drążka może być odczytywana za pomocą instrukcji:

INKEY $

Wynika stąd pewna niedogodność: wyrażenie INKEY $ będzie jednocześnie przyjmowało wartości przyporządkowane tylko jednemu klawiszowi, a więc np., nie będzie możliwe przesłanie polecenia "w górę" i "strzal" równocześnie.

Alternatywnym — i bardziej sensownym — sposobem tworzenia programów z użyciem joysticków będzie użycie funkcji:

IN 61438 (odczytuje pozycje drążka I)

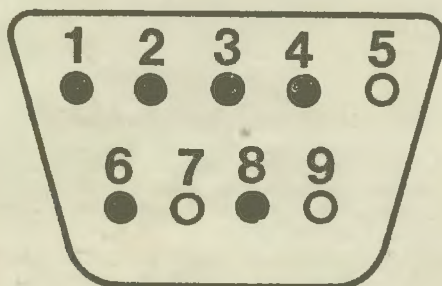
IN 63486 (odczytuje pozycje drążka II)

Przedstawiony poniżej program pozwala na rysowanie na ekranie monitora dowolnych kształtów poprzez odpowiednie poruszanie drążkiem. Naciśnięcie przycisku dodatkowego powoduje zniknięcie rysującego punktu. Będzie się on poruszał nadal, tak jak dotychczas, a ukaże się w chwili zwolnienia przycisku.

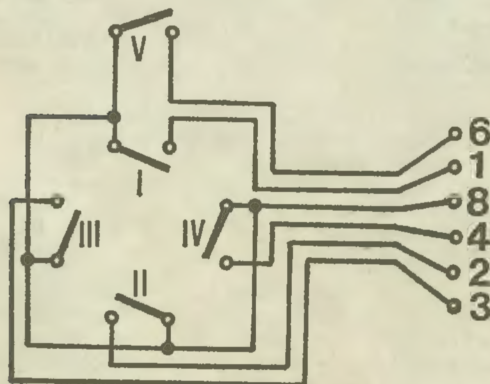
```
1Ø LET x=Ø: LET y=8Ø
2Ø GOSUB 1ØØØ
3Ø IF rys=1 THEN PLOT x,y
4Ø GO TO 2Ø
1ØØØ LET a=IN 61438
1Ø1Ø IF a>127 THEN LET a=a—128
1Ø2Ø IF a>63 THEN LET a=a—64
1Ø3Ø IF a>31 THEN LET a=a—32
1Ø4Ø IF a>15 THEN LET a=a—16:IF x<255 THEN LET x=x+1
1Ø5Ø IF a>7 THEN LET a=a—8:IF x>Ø THEN LET x=x—1
1Ø6Ø IF a>3 THEN LET a=a—4:IF y<175 THEN LET y=y+1
1Ø7Ø IF a>1 THEN LET a=a—2:IF y>Ø THEN LET y=y—1
1Ø8Ø LET rys=a
1Ø9Ø RETURN
```

O tym, jakie wartości przyjmują wyrażenia IN 61438 oraz IN 63486 możecie się przekonać sami, analizując działanie podanego programu. Możecie też zapytać o to Waszego komputera — wystarczą trzy instrukcje.

ROMAN POZNAŃSKI



Rys. 5. Standardowy wtyk połączeniowy



Rys. 6. Schemat elektryczny układu drążka

OCHRONA PROGRAMÓW

Ochrona programów przed nieautoryzowanym kopiowaniem i rozpowszechnianiem to w głównej mierze problem firm software'owych. Bywa jednak, że i indywidualni użytkownicy komputerów stają przed koniecznością wprowadzenia takich zabezpieczeń do własnych programów.

W przypadku komputera ZX SPECTRUM jest to sprawa szczególnie skomplikowana. Prostota systemu operacyjnego i jego konstrukcja z jednej strony daje dużą łatwość obsługi, z drugiej jednak pozbawia możliwości zakładania efektywnych zabezpieczeń. W konsekwencji każdy program napisany na SPECTRUM może być „złamany" to znaczy wydrukowany, przeanalizowany i skopiowany.

W praktyce odbywa się to w następujący sposób. Pożycza się od kogoś kasetę z interesującym programem, ładuje do komputera program kopiujący i po kilku minutach ma się na taśmie własny egzemplarz identycznie zabezpieczony jak oryginał. W niewielu przypadkach powyższa metoda zawodzi.

Poniżej przedstawiam sposób na włączenie Twoich programów do grupy tych nielicznych, które nie poddają się programom kopiującym. Proponowane zabezpieczenie, choć proste w swej idei, do złamania wymaga niezłej znajomości kodu maszynowego oraz zasad działania ROM-u. Jest więc całkowicie skuteczne wobec wszystkich tych, którzy bazują jedynie na programach kopiujących i co najwyżej potrafią obezwładniać proste blokady wmontowane w programy BASIC-owe. Na fachowców niestety nie ma żadnej rady. Tym można co najwyżej zabrać więcej czasu. Jest ich na szczęście (?) niewielu i na ogół mają ciekawsze zajęcia niż łamanie Twoich programów.

Przystąpmy więc do pracy. Wprowadź do komputera poniższy program i uruchom go instrukcją RUN.

```
1Ø REM 1ØØ dowolnych symboli
2Ø FOR i=Ø TO 9: LET s=Ø
3Ø FOR j=1 TO1Ø
4Ø READ a: LET s=s+a:
5Ø POKE 23758+1Ø×i+j,a
6Ø NEXT j:READ a
7Ø IF a=s THEN PRINT "Linia nr "; 2ØØ+1Ø×i;" O.K.";GO TO 9Ø
8Ø PRINT "Linia nr ";2ØØ+1Ø×i;" wymaga poprawy.": STOP
9Ø NEXT i
1ØØ RANDOMIZE USR 23759
11Ø SAVE "na tasme" CODE 65368.134
12Ø PRINT "Przewin kasete· celem weryfikacji"
13Ø VERIFY ""CODE 65368,134
2ØØ DATA 33,194.4,17,146,255,1,76,Ø,237,963
21Ø DATA 176.33,168,255,34,182,255,33,213,255,16Ø4
22Ø DATA 34,2Ø4 255,33,218.255,54,167,33,247,15ØØ
23Ø DATA 92.17.83.255,1,57.Ø.237.176.2Ø1,1124
24Ø DATA 237,75,83,92,42,89.92,175,237,66,1188
25Ø DATA 34,137,255,42,75,92,237,66,34,14Ø,1112
26Ø DATA 255,221,33,135,255,17,17,Ø,2Ø5,194,1332
27Ø DATA 4,42,83,92,34,137,255,33,255,255,119Ø
28Ø DATA 43,124,181,32,251,62.24Ø,221,33,Ø,1187
29Ø DATA Ø,17,Ø,Ø,24,2,Ø,Ø,Ø,Ø,43
```

(W linii 10 po instrukcji REM umieść naprawdę 100 symboli a nie jedynie ten krotki napis!) Jeżeli pierwszy przebieg programu wykaże, że któraś z linii zawierających dane jest błędna, to możesz mieć kłopoty z wypisaniem tekstu programu na ekranie. Użyj wtedy komendy LIST 20 i nie przejmuj się dziwaczną linią 10. Podobnie, kolejnych uruchomień programu dokonuj instrukcją RUN 20 lub GO TO 20.

Teraz usuń linię 10 oraz wszystkie linie o numerach 100 i większych. Zamiast nich wprowadź do komputera poniższe:

```
1 REM 110 symboli
2ØØ DATA 33,115,8,17,88,255,1,63,Ø,237,817
21Ø DATA 176,221,33,88,255,221,54,11.2,42,11Ø3
22Ø DATA 61,92,221.54,14,3,54,Ø,221.54,774
23Ø DATA 32,5,35,221,54,35,6,54,Ø,221,663
24Ø DATA 54,42,1Ø,221,54,5Ø,9,221,54.63,778
25Ø DATA 7,33,86,5,17,152,255,1,82,Ø,638
26Ø DATA 237,176.33,46.32,34,233,255,31,195,1274
27Ø DATA 2Ø2,34,235,255,33,5,Ø,34,237,255,129Ø
28Ø DATA 221,33,Ø,91,17,17,Ø,62,Ø,55,496
29Ø DATA 221,229,2Ø5,86,5,221.225,42,83,92,14Ø9
3ØØ DATA 195,88,255,Ø,Ø,Ø,Ø,Ø,Ø,Ø,538
```

Ponownie uruchom program przez RUN. Po poprawnym zakończeniu usuń wszystkie linie oprócz pierwszej i nagraj ją na kasetę komendą SAVE „z tasmy".

Jeśli nie popełniłeś błędu to jesteś już gotów do zabezpieczenia każdego programu napisanego w BASIC-u.

Najpierw musisz napisać króciutki program ładujący, którego celem będzie wczytanie do komputera Twojego właściwego programu i ewentualnie miłego dla oka zapełnienia ekranu na czas ładowania. Zamiast instrukcji LOAD "nazwa" musisz jednak napisać **RANDOMIZE USR 65368**. W najprostszym przypadku będzie to wyglądać tak:

10 RANDOMIZE USR 65368. Następnie wczytaj z taśmy instrukcją **MERGE "z tasmy"** ostatnio skonstruowany blok. Przygotuj kasetę, na której ma być umieszczona zabezpieczona kopia Twojego dzieła i nagraj na nią program ładujący rozkazem **SAVE "nazwa" LINE 10.**

Wczytaj do komputera blok kodu maszynowego "na tasme" a następnie własny program do zabezpieczenia. Nagraj go na przygotowaną kasetę zaraz za programem ładującym rozkazem RANDOMIZE USR 65368. Pamiętaj, że nagrywanie rozpocznie się natychmiast po wciśnięciu klawisza ENTER. Jeśli Twój program wczytuje jeszcze jakieś bloki (np. symbole graficzne definiowane przez użytkownika) to nagraj je na końcu. Program jest już bezpieczny.

Na koniec kilka uwag praktycznych. Zanim zabezpieczysz swój program upewnij się, że nie zawiera on żadnych błędów, gdyż w razie wystąpienia jakiegokolwiek, natychmiast sam się skasuje. To samo nastąpi przy każdej próbie przerwania go w trakcie wykonywania się jak również w razie wystąpienia błędu podczas wczytywania z kasety do komputera.

Po wczytaniu, program będzie się automatycznie uruchamiał od pierwszej linii. Jeśli Ci to nie odpowiada to przed nagraniem go na taśmę wykonaj z klawiatury rozkazy:

POKE 65464,m—256×INT(m/256)
POKE 65465,INT(m/256)

gdzie m jest numerem linii startowej.

Jeśli znasz inne sposoby ochrony programów przed „ciekawskimi" to śmiało je stosuj równolegle z powyższym. Prawdopodobieństwo „tragicznej w skutkach kolizji" jest niewielkie. Im więcej stworzysz barier tym większa szansa, że zniechęcisz potencjalnego pirata do zmagań z Twoim programem. Szczególnie warto popracować nad programem ładującym, by maksymalnie utrudnić jego rozpracowanie. Również i w głównym programie warto umieścić kilka pułapek, mimo że jest on już dosyć skutecznie chroniony.

ANDRZEJ KADLOF

Największa przyjemność

DOKOŃCZENIE ZE STR. 32

Teraz nasz komputer już wie, że ma określić A i B, każdą z tych zażądanych przez nas liczb w przedziale od 0 do 10. Powinien teraz napisać nam na ekranie telewizora, co „wymyślił". A ponieważ dzisiejsza klasówka jest z dodawania każemy mu to napisać w formie najprostszego równania. Programujemy:

3∅ PRINT A; "+"; B; "="

(PRINT znaczy napisz)

oraz klawisz ENTER. Mamy już na ekranie napisane „wymyślone" przez naszego egzaminatora równanie. Najpierw musi oczywiście rozwiązać je sam, a potem zapytać o nasz wynik. Rozwiązanie komputera jest proste, zapisujemy je tak:

4∅ LET C = A + B i ENTER

Teraz kolej na pytanie o wynik obliczony przez nas:

5∅ INPUT D i ENTER

Po tym zadanym przez komputer pytaniu czeka on z dalszą realizacją programu do chwili, aż podamy mu wynik naszych obliczeń (naciskając odpowiednie klawisze) i na koniec oczywiście ENTER. Trzeba teraz, aby nasza (miejmy nadzieję, że właściwa odpowiedź) została w odpowiednim miejscu już napisanego równania, wydrukowana. Służy temu instrukcja:

6∅ **PRINT** (teraz to właściwe miejsce) **AT ∅, 1∅;** D i ENTER

Czas na odpowiedź naszego egzaminatora. Jeżeli nasz wynik będzie niewłaściwy, zostanie to skwitowane krótkiem „ŹLE", z tym, że ze względu na brak odpowiedniej litery zamiast Ź będzie Z. Dobre rozwiązanie spowoduje uznanie ze strony komputera: „BRAWO".

Wystarczy, że dopiszemy jeszcze instrukcje:

7∅ IF D = C THEN "BRAWO"

i znane już ENTER. Wreszcie:

8∅ IF D<>C THEN PRINT „ZLE"

i ENTER. I oto nasz mini program jest gotowy. Po wpisaniu w pamięć komputera na rozkaz LIST zostanie wydrukowany w całości i będzie wyglądał tak:

```
1∅ LET A = INT (RND × 11)
2∅ LET B = INT (RND × 11)
3∅ PRINT A; "+"; B "="
4∅ LET C = A + B
5∅ INPUT D
6∅ PRINT AT ∅, 1∅; D
7∅ IF D = C THEN PRINT "BRAWO"
8∅ IF D<>C THEN PRINT „ZLE"
```

Pozostaje nam tylko nacisnąć RUN i przystępujemy do egzaminu.

Po każdej odpowiedzi, kiedy już wiemy czy jest dobra, wciskamy RUN oraz ENTER i zabawa trwa. Jeżeli zaś już opanowaliśmy liczenie przy A i B od zera do dziesięciu, możemy sami określić, w jakich granicach ma wybierać komputer A i B. Wystarczy zmienić:

INT (RND × 11)

na przykład

INT (RND × 21)

i będziemy rozwiązywać dodawania z liczb od 0 do 20. W ten sposób, przez samodzielne programowanie sami wyznaczamy sobie coraz to inne zadania i to właśnie jest ta NAJWIĘKSZA PRZYJEMNOŚĆ.

MAREK

W następnym numerze m.in.:

- Wywiad ze Steve Wozniakiem
- Kilka uwag o algorytmach
- Mikro-PROLOG
- Informatyka na „Perskim"
- Pióro świetlne
- Commodore C-128
- Roger Zelazny — Sekskomputer

Największa przyjemność

No, nareszcie! Dziś w naszej rodzinie wielkie święto, od dawna oczekiwane i wytęsknione. Nareszcie w naszym domu pojawił się On i od razu życie zaczęło płynąć innym nurtem niż dotychczas. Objętościowo nie jest na szczęście zbyt duży, zajmuje tylko kawałek półki w sąsiedztwie telewizora, ale prawdopodobnie w przyszłości dorosną do Niego rozmaite inne urządzenia. Zaraz po przyjściu z pracy opędzamy byle szybko najniezbędniejsze czynności i w pośpiechu sadowimy się przed ekranem telewizora, mając na podorędziu niezbędne kanapki i coś do picia. Z westchnieniem ulgi wkładamy kasetę do magnetofonu i za chwilę jesteśmy już panami sytuacji, władcami rozległych krain i różnorodnego sprzętu technicznego posłusznego naszym palcom. Na ekranie sytuacja zmienia się w olbrzymim tempie, wyrastają coraz to nowe przeszkody, wskaźniki i mapy, a my z trudem nadążamy (albo i nie) za rozwojem akcji. Od czasu do czasu dopuszczamy do magicznej skrzyneczki nasze dziecko, wyraźnie po pewnym czasie znudzone obserwowaniem naszych zmagań z programem kolejnej gry

Przedszkolaki podnieście bunt!

Wystarczy przecież kilka minut, aby KOMPUTER z groźnego i strzelającego przeciwnika stał się Waszym przyjacielem, który na przykład przeegzaminuje Waszą znajomość matematyki, będzie przy tym wyrozumiały dla popełnianych błędów i nigdy się nie zmęczy. Na początek powiem Wam, jak można żądać od komputera, aby sprawdził, czy już umiecie dodawać dwie liczby, z których każda nie jest większa od 10.

Najpierw trzeba tak zaprogramować komputer, żeby sam wymyślił dwie liczby, które będziemy dodawać. (Oczywiście komputer nie myśli, ale dla uproszczenia nazwijmy tak tę czynność). Jedną z tych liczb nazwijmy A, a drugą B. Nasz komputer „wymyśli" je, gdy podamy mu instrukcję:

LET A = INT (RND × 11)
LET znaczy niech

To samo dotyczyć będzie określenia B. Należy teraz wpisać tę instrukcję do programu, który komputer zrealizuje.

Przy wpisywaniu instrukcji do programu istnieje bardzo pożyteczny zwyczaj numerowania jej co dziesięć, umożliwia to wpisywanie dodatkowych zleceń dla komputera. Tak więc napiszmy:

10 LET A = INT (RND × 11)

i naciskamy klawisz ENTER, co komputer rozumie jako koniec tego rozkazu i polecenie zapisania go w swojej pamięci. Określamy teraz B:

2Ø LET B = INT (RND × 11) i ENTER

DOKOŃCZENIE NA STR. 31

